Received: 9 October 2019 | Revised: 1 April 2020 | Accepted: 26 May 2020

DOI: 10.1002/nme.6447

RESEARCH ARTICLE

WILEY

Parametrized reduced order modeling for cracked solids

Konstantinos Agathos[1] | **Stéphane P. A. Bordas**[2,3,4] | **Eleni Chatzi**[1]

[1]Department of Civil, Environmental, and Geomatic Engineering, ETH Zürich, Zürich, Switzerland

[2]Research Unit in Engineering Science, Luxembourg University, Luxembourg, Luxembourg

[3]Institute of Mechanics and Advanced Materials, School of Engineering, Cardiff University, UK

[4]Department of Medical Research, China Medical University Hospital, China Medical University, Taichung, Taiwan

Correspondence
Konstantinos Agathos, Department of Civil, Environmental, and Geomatic Engineering, ETH Zürich, Zürich CH-8093, Switzerland.
Email: agathos@ibk.baug.ethz.ch

Summary

A parametrized reduced order modeling methodology for cracked two dimensional solids is presented, where the parameters correspond to geometric properties of the crack, such as location and size. The method follows the offline-online paradigm, where in the offline, training phase, solutions are obtained for a set of parameter values, corresponding to specific crack configurations and a basis for a lower dimensional solution space is created. Then in the online phase, this basis is used to obtain solutions for configurations that do not lie in the training set. The use of the same basis for different crack geometries is rendered possible by defining a reference configuration and employing mesh morphing to map the reference to different target configurations. To enable the application to complex geometries, a mesh morphing technique is introduced, based on inverse distance weighting, which increases computational efficiency and allows for special treatment of boundaries. Applications in linear elastic fracture mechanics are considered, with the extended finite element method being used to represent discontinuous and asymptotic fields.

KEYWORDS

ECSW, fracture, IDW, mesh morphing, model order reduction, XFEM

1 | INTRODUCTION

In several engineering applications, such as design optimization and damage detection, the repeated solution of fracture mechanics problems for different crack configurations is required. The solution of such problems can be facilitated by the use of enriched finite element methods,[1-3] which allow to represent cracks independently of the finite element mesh used. In the field of damage detection in particular, the use of the extended finite element method (XFEM)[1,2] in conjunction to optimization algorithms, initially proposed for the detection of cracks in two-dimensional solids under static loading,[4] has become quite common. Thus, several extensions can be found in the literature, which pertain to dynamic loading,[5,6] detection of holes,[7,8] multiple flaws,[9-11] and three dimensional solids.[12,13] Despite the success of the above schemes, a drawback that could limit their applicability to realistic problems is the relatively high computational cost associated with the solution of a large number of static or dynamic problems involving different crack or void configurations. In this context, parametrized model order reduction (MOR) could be employed as a means for decreasing the required computational time and eventually allowing for the online application of such methods, thus greatly extending their applicability.

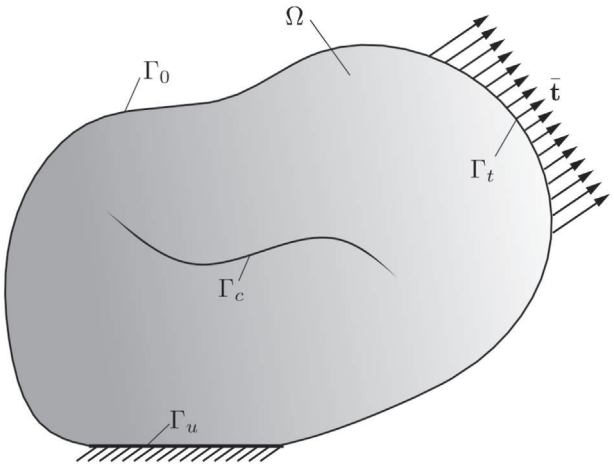
MOR collectively refers to a set of techniques used to accelerate the solution of problems, typically described by large and computationally expensive numerical models, termed full order models (FOM). A common setting for the application of such methods is the repeated solution of parameter dependent problems for different values of the parameters. From the several categories of MOR techniques, projection based methods are the most relevant to the present work. These methods have been extensively applied to a wide array problems in different fields of engineering such as structural dynamics and fluid mechanics, allowing to obtain similar accuracy as their full order counterparts at a fraction of the numerical cost. Projection based MOR follows the offline-online paradigm, where a set of computationally expensive pre computations are performed offline to construct the reduced order model (ROM), which can be efficiently solved online. The offline phase consists of constructing a basis for a subspace of lower dimensionality than the FOM, where the solution will be sought for. This basis can be constructed using different methods, with the most popular choice being the proper orthogonal decomposition (POD) combined with the method of snapshots.[14,15] Within the POD method, parameter dependence can be introduced either through the construction of a global basis, able to represent the solution over a range of parameter values, or through the construction of multiple local bases. In the former case, snapshots of solutions for different parameter values should be obtained. In the latter, some additional technique, such as interpolation in the tangent space of an appropriate manifold[16] or interpolation of the angles between subspaces,[17] needs to be used to obtain bases for non-sampled values of the parameters. Once a basis has been constructed, the discretized equations of the system are projected on this space, thus effectively reducing the problem size. For linear systems and specific types of parameter dependence, the projection on the lower dimensional subspace can be pre computed in the offline phase, leaving only the solution of the reduced problem for the online phase. An alternative for linear or linearized problems, and general parameter dependence, consists of precomputing the projections for specific parameter values and employing some kind of interpolation for intermediate values.[18-21] For nonlinear problems and general parameter dependence, special methods, also known as hyper reduction methods,[22] are employed to allow the fast computation of the required projections in the online phase.

In the field of fracture mechanics, several works can be found that employ, in various forms, projection based MOR for problems involving static cracks. In the work of Huynh and Patera,[23] stress intensity factors (SIFs) of cracked specimens are predicted using projection based ROMs, parametrized with respect to the crack length and other material and geometrical properties. Dependence on the crack length was handled by introducing a reference geometry to which different configurations were mapped, while XFEM was used for the corresponding FOMs. Niroomandi et al[24] proposed a technique, based on XFEM, to simulate cutting for surgery simulations. Different positions and geometries for the cuts could be handled by superimposing a patch of enriched elements on a set of global modes, obtained for the uncut structure. Of course, this technique can only account for local effects of the cut. Similarly, O'Hara and Hollkamp[25] combine global ROMs to refined local models, able to represent the crack, through a one way coupling approach. In that work, crack propagation was also modeled. Wang et al[26,27] locally enrich modes obtained for healthy structures to account for cracks with local effects. In a recent work, He et al[28] introduce a local modification to projection based ROMs for linear elastic fracture mechanics in order to improve the representation of singular fields. This work however did not allow for any parameter variation. For crack propagation problems, Galland et al[29] proposed an MOR methodology based on mesh morphing to model crack propagation in three dimensional solids. Finally, Kerfriden[30-32] employ a series of techniques, such as the use of Krylov subspaces, enrichment of the reduced basis and domain decomposition to reduce the computational cost associated with crack propagation problems.

The techniques introduced in the present work enable the use of projection based MOR for problems involving cracks, while using as parameters geometric properties of the cracks. The main goal is to allow the efficient solution of such problems for a wide range of crack configurations by exploiting information from a finite set of representative cases solved during an offline training phase. Moreover, in contrast to some of the previously mentioned works, the accuracy of the resulting models is independent of the crack size. Nevertheless, it should be noted that since these methods rely on parametrizing the crack geometry, they are mainly applicable to problems involving static cracks or limited crack propagation, where the number of parameters is relatively small. The above features render the techniques proposed herein suitable for use in crack detection problems, which is the target application, and also other fields such as design optimization or prognosis.

The remainder of the article is organized as follows. In Section 2, the linear elasticity problem in the presence of cracks, as well as the discretization scheme used are presented. In Section 3, the handling of discontinuities and singularities in the framework of projection based MOR is discussed and the use of the proposed strategy is justified. In Section 4, the proposed mesh morphing method is presented. In Section 5.2.1, the techniques used to obtain lower dimensional bases and accelerate the online performance of the method are briefly described. In Section 6, different aspects of the method

FIGURE 1 Cracked solid with boundary conditions



are tested through several numerical problems. Finally, in Section 7, some the most important results are summarized and possible directions for future work are set.

2 | PROBLEM STATEMENT AND FOM

The main focus in the present work is on the construction of ROMs for cracked solids, allowing to approximate the solution for different crack configurations. Thus, although the methods presented are applicable to dynamic and, subject to certain limitations, to nonlinear and three dimensional problems, in the following, we will only consider the case of cracked linear elastic solids under static loading.

2.1 | Weak form

The model problem is formulated for a general cracked domain Ω, as shown in Figure 1. The domain boundary is decomposed as $\Gamma{=}\Gamma_0{\cup}\Gamma_u{\cup}\Gamma_t{\cup}\Gamma_c$, with Γ_0 designating the part where no boundary conditions are applied, Γ_u the part where Dirichlet conditions are applied in the form of prescribed displacements $\bar{u}$, and Γ_t denotes the part where Neumann conditions are applied in the form of surface tractions $\bar{\boldsymbol{t}}$. In the crack surface Γ_c, free boundary conditions apply.

Then, the elastostatics problem can be written in weak form as:

Find $\boldsymbol{u} \in \mathcal{U}$ such that $\forall \boldsymbol{v} \in \mathcal{V}^0$

$$\int_\Omega \boldsymbol{\sigma}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{v})\, d\Omega = \int_\Omega \boldsymbol{b} \cdot \boldsymbol{v}\, d\Omega + \int_{\Gamma_t} \bar{\boldsymbol{t}} \cdot \boldsymbol{v}\, d\Gamma, \tag{1}$$

where:

$$\mathcal{U} = \{\boldsymbol{u} | \boldsymbol{u} \in (H^1(\Omega))^3, \boldsymbol{u} = \bar{\boldsymbol{u}} \text{ on } \Gamma_u\}, \tag{2}$$

and

$$\mathcal{V}^0 = \{\boldsymbol{v} | \boldsymbol{v} \in (H^1(\Omega))^3, \boldsymbol{v} = 0 \text{ on } \Gamma_u\}. \tag{3}$$

It should be noted that $\boldsymbol{u}$ and $\boldsymbol{v}$ are discontinuous along Γ_c.

In Equation (1), $\boldsymbol{\sigma}$ is the Cauchy stress tensor, and $\boldsymbol{b}$ is the applied body force per unit volume. By assuming linear elastic material behavior, stresses can be obtained from strains through Hooke's law:

$$\boldsymbol{\sigma} = \boldsymbol{D} : \boldsymbol{\varepsilon}. \tag{4}$$

The weak form can thus be recast as:

Find $\boldsymbol{u} \in \mathcal{U}$ such that $\forall \boldsymbol{v} \in \mathcal{V}^0$:

$$\int_\Omega \boldsymbol{\varepsilon}(\boldsymbol{u}) : \boldsymbol{D} : \boldsymbol{\varepsilon}(\boldsymbol{v})\, d\Omega = \int_\Omega \boldsymbol{b} \cdot \boldsymbol{v}\, d\Omega + \int_{\Gamma_t} \bar{\boldsymbol{t}} \cdot \boldsymbol{v}\, d\Gamma. \tag{5}$$

If small deformations are further assumed, the strain field $\boldsymbol{\varepsilon}$ can be described as as the symmetric gradient of the displacement field:

$$\boldsymbol{\varepsilon} = \boldsymbol{\nabla}_s \boldsymbol{u}. \tag{6}$$

2.2 | Full order model

From the several alternatives available for discretizing the problem described in the previous subsection, we will employ the extended/generalized finite element method[1-3] (XFEM/GFEM). This choice is mainly motivated by the ability of these methods to accurately represent singular fields present in the solution through partition of unity enrichment. Additionally, in these methods, discontinuities are represented independently of the underlying finite element mesh. Thus, different crack configurations can be treated without requiring remeshing, a feature that, as will be discussed in what follows, can facilitate the automated generation of ROMs covering a wide range of possible crack configurations.

2.2.1 | Crack representation

Within the XFEM/GFEM framework, surfaces, along which discontinuities occur, are most commonly described implicitly using signed distance or level set functions.[33,34] For open surfaces, such as cracks, two functions are required:

- The normal level set ϕ, defined as the signed distance from the crack surface:

$$\phi(\mathbf{x}) = \min_{\bar{\mathbf{x}} \in \Gamma_c} \|\mathbf{x} - \bar{\mathbf{x}}\| \text{sign}(\mathbf{n}^+ \cdot (\mathbf{x} - \bar{\mathbf{x}})), \tag{7}$$

  where $\mathbf{n}^+$ is the outward normal to the crack surface and sign(□) is the sign function.
- The tangential level set ψ, defined as the signed distance function satisfying the conditions:

$$\nabla\phi \cdot \nabla\psi = 0 \tag{8a}$$

$$\left.\begin{aligned} \phi(\mathbf{x}) &= 0 \\ \psi(\mathbf{x}) &= 0 \end{aligned}\right\} \forall \boldsymbol{x} \in \Gamma_f, \tag{8b}$$

  where Γ_f are the crack front/tips.

2.2.2 | Enrichment functions

To allow for the representation of discontinuities, the finite element approximation is enriched with the modified Heaviside function:

$$H(\phi) = \begin{cases} 1, & \text{for } \phi \geq 0 \\ -1, & \text{for } \phi < 0 \end{cases}. \tag{9}$$

In order to represent the stress singularities at the crack tips/front, a polar system with its origin at these locations is employed. The coordinates of this system can be conveniently defined with the aid of the level set functions:

$$r = \sqrt{\phi^2 + \psi^2}, \quad \theta = \arctan\left(\frac{\phi}{\psi}\right). \tag{10}$$

Using the above, the set of asymptotic enrichment functions introduced in the work of Belytschko and Black[1] can be defined:

$$F_j(r,\theta) = \left\{ \sqrt{r}\sin\frac{\theta}{2}, \sqrt{r}\cos\frac{\theta}{2}, \sqrt{r}\sin\frac{\theta}{2}\sin\theta, \sqrt{r}\cos\frac{\theta}{2}\sin\theta \right\}. \tag{11}$$

2.2.3 | Displacement approximation

Combining the previously defined enrichment functions, the displacement approximation for the standard XFEM with shifted enrichment functions[35] assumes the form:

$$\mathbf{u}(\mathbf{x}) = \sum_{I\in\mathcal{N}} N_I(\mathbf{x})\mathbf{u}_I + \sum_{J\in\mathcal{N}^j} N_J(\mathbf{x})(H(\mathbf{x}) - H(\boldsymbol{x}_J))\mathbf{b}_J + \sum_{T\in\mathcal{N}^t}\sum_{j} N_T(\mathbf{x})(F_j(\mathbf{x}) - F_j(\boldsymbol{x}_T))\mathbf{c}_{Tj}, \tag{12}$$

where N_I are the FE shape functions, $\mathbf{u}_I$ are nodal displacements, $\boldsymbol{x}_J$ are nodal point coordinates, and $\mathbf{b}_J$ and $\mathbf{c}_{TJ}$ are enriched degrees of freedom.

The nodal sets of Equation (12) can be defined with the aid of the level set functions:

$\mathcal{N}$ is the set of all nodes in the FE mesh.

$\mathcal{N}^j$ is the set of discontinuity enriched nodes. This nodal set includes all nodes belonging to elements where the first level set changes sign.

$\mathcal{N}^t$ is the set of asymptotically enriched nodes. This nodal set includes all nodes belonging to elements where both level sets change sign.

The enrichment scheme described above, often termed "topological enrichment," does not recover the optimal convergence rate of the underlying FE approximation, despite the fact that it employs singular enrichment. As an alternative, which allows to recover optimal convergence rates, "geometrical enrichment"[36,37] can be employed, where all nodes within a predefined distance from the singularity are enriched using the functions of Equation (11). In the remainder of this work, the "topological enrichment" scheme will be used due to its simplicity.

3 | POD FOR DISCONTINUITIES/SINGULARITIES

As already mentioned, the main goal of the present work is to allow the construction of parametrized ROMs for cracked solids, where the parameters geometrically describe the crack. Ideally, these models should be trained using solutions from specific realizations of the model in an offline phase. Then in the online phase, they should allow to approximate the solution of the FOM at a fraction of its computational cost for any value of the parameters within some range.

In the following, we briefly describe how the above is realized in the general case of parametrized reduced order modeling. Next, we highlight some of the challenges involved when specializing the methodology to the case of cracks or discontinuities in general. Finally, we discuss some possibilities for overcoming the aforementioned challenges.

3.1 | Projection based reduced order modeling

Discretization of the weak form of Equation (5) with the approximation of Equation (12) results in a linear system of equations of the form:

$$\boldsymbol{K}(\boldsymbol{\mu})\hat{\boldsymbol{u}}(\boldsymbol{\mu}) = \boldsymbol{f}(\boldsymbol{\mu}), \tag{13}$$

where $\boldsymbol{K} \in \mathbb{R}^{N\times N}$ is the stiffness matrix, $\overline{\boldsymbol{u}} \in \mathbb{R}^N$ is the vector of nodal displacements, $\boldsymbol{f} \in \mathbb{R}^N$ is the force vector and $\boldsymbol{\mu} \in \mathbb{R}^{N_p}$ is the vector of parameters, N is the number of nodal unknowns, including enriched dofs in the case of enriched approximations, and N_p is the number of parameters.

The problem size can be drastically reduced by seeking for an approximate solution in a lower dimensional space of size $k \ll N$. Then the approximate solution can be written as:

$$\hat{\boldsymbol{u}}(\boldsymbol{\mu}) \approx \boldsymbol{V}(\boldsymbol{\mu})\boldsymbol{y}, \tag{14}$$

where $\boldsymbol{V} \in \mathbb{R}^{N\times k}$ is a set of basis vectors of the lower dimensional space and $\boldsymbol{y} \in \mathbb{R}^k$ a vector of coordinates in this space. By substituting Equation (14) in the system of Equation (13) and further projecting forces on the lower dimensional space, the reduced equilibrium equations can be obtained:

$$\boldsymbol{V}^T(\boldsymbol{\mu})\boldsymbol{K}(\boldsymbol{\mu})\boldsymbol{V}(\boldsymbol{\mu})\boldsymbol{y} = \boldsymbol{V}(\boldsymbol{\mu})^T\boldsymbol{f}(\boldsymbol{\mu}), \tag{15}$$

which can be written compactly as:

$$\tilde{\boldsymbol{K}}(\boldsymbol{\mu})\boldsymbol{y} = \tilde{\boldsymbol{f}}(\boldsymbol{\mu}) \tag{16a}$$

$$\tilde{\boldsymbol{K}}(\boldsymbol{\mu}) = \boldsymbol{V}(\boldsymbol{\mu})^T\boldsymbol{K}(\boldsymbol{\mu})\boldsymbol{V}(\boldsymbol{\mu}) \tag{16b}$$

$$\tilde{\boldsymbol{f}}(\boldsymbol{\mu}) = \boldsymbol{V}(\boldsymbol{\mu})^T\boldsymbol{f}(\boldsymbol{\mu}), \tag{16c}$$

where $\tilde{\boldsymbol{K}} \in \mathbb{R}^{k\times k}$ is the reduced stiffness matrix and $\tilde{\boldsymbol{f}} \in \mathbb{R}^k$ is the reduced external force vector.

For the construction of the lower dimensional basis, different possibilities exist regarding both handling of the parameter dependence and construction of the basis for a single set of parameters. As already mentioned, parameter dependence can be taken into account either by constructing a number of local bases, each one corresponding to a specific set of parameter values, and employing interpolation for intermediate values,[16] or by using a single global basis covering the whole parameter range. Herein, the latter approach is used due to the relatively small size of the bases required for a single set of parameter values. Thus, in the following, the dependence of the lower dimensional basis of Equation 14 on the parameter vector $\boldsymbol{\mu}$ is dropped. Regarding construction of the basis, we employ the POD.[14] To this end, the high-dimensional problem is solved for a set of representative values of the parameters and the solutions, also termed snapshots, are stored in a matrix of the form:

$$\overline{\boldsymbol{S}} = \left[\hat{\boldsymbol{u}}(\boldsymbol{\mu}_1) \quad \hat{\boldsymbol{u}}(\boldsymbol{\mu}_2) \quad \dots \quad \hat{\boldsymbol{u}}(\boldsymbol{\mu}_{N_s})\right], \tag{17}$$

where $\hat{\boldsymbol{u}}(\boldsymbol{\mu}_i)$ are the snapshots, $\boldsymbol{\mu}_i$ are the selected parameter values and N_s is the number of snapshots. Then, the reduced basis can be formed by performing a singular value decomposition (SVD) on the snapshot matrix:

$$\overline{\boldsymbol{S}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{Z}^T, \tag{18}$$

and selecting the first k columns of $\boldsymbol{U}$:

$$\boldsymbol{V} = \left[\boldsymbol{U}_1 \quad \boldsymbol{U}_2 \quad \cdots \quad \boldsymbol{U}_k\right], \tag{19}$$

with $\boldsymbol{U}_i$ the ith column of $\boldsymbol{U}$.

In this process, the selection of snapshots is critical for the accuracy of the resulting ROM over the whole parameter range. The specific techniques employed in the present work are discussed in Section 5.1

3.2 | POD for cracks

Direct application of the above procedure to problems where the parameters $\boldsymbol{\mu}$ describe the location of cracks or discontinuities would result in highly inaccurate modes. More specifically, since in the computed snapshots the discontinuities

would lie at different locations, the modes, as a linear combination of the snapshot solutions, would essentially include several discontinuities. Based also on existing literature, four possible approaches have been identified for overcoming the above problem, as described in what follows.

3.2.1 | Decomposing the solution into local and global parts

In this approach, it is assumed that the crack only locally affects the solution, thus the modes of the healthy solid can be used for the largest part of the structure. To represent the local behavior in the vicinity of the crack, either the corresponding part of the full model is retained,[24] or the lower dimensional space is enriched with modes representing the discontinuity.[26,27] These approaches offer increased flexibility since, once a reduced model has been constructed for the healthy structure, it can be modified online to account for any possible crack location as well as crack propagation. The main limitation however, is that the crack should be small enough to leave the healthy modes of the structure unaffected.

3.2.2 | Mesh morphing

This approach consists of building a FOM for a given crack configuration and subsequently modifying it so that it can be used for different, but similar crack configurations. This modification can be realized through mesh morphing, that is, the process of altering the geometry of a mesh by modifying the locations of the nodes while preserving element connectivities. Then, an ROM can be constructed based on the initial parametrized FOM. This technique has been applied to problems with static cracks in the work of Huynh and Patera[23] and to crack propagation problems in three dimensional solids in the work of Galland et al.[29] In the latter, a reduced basis was built for an initial crack and subsequently mesh morphing was employed to match the original mesh to the propagated crack, while solutions of the full problem were performed to enrich the lower dimensional subspace when errors would exceed a certain threshold. Compared to the previous alternative, this technique offers the advantage of allowing to exploit information obtained for different crack configurations. The main limiting factor of this approach is the mesh morphing process itself, which for large discrepancies between the initial and target crack can lead to excessive mesh distortion, resulting in large errors or even failure of the solution process.

3.2.3 | Mapping the modes

As an alternative to mesh morphing, a fixed mesh could be employed in conjunction to mapping of the snapshots and modes. More specifically, the process would consist of defining a reference configuration and mapping all the snapshots to this configuration. Subsequently, the lower dimensional subspace could be computed in the reference configuration and mapped to each target configuration during the online phase. Mapping of the snapshots to the reference configuration would require some method to map the geometry used for the snapshots to the reference one as well as interpolation. Mapping of the modes from the reference to each target configuration would require evaluation of the inverse of the aforementioned mapping and again interpolation. Some works implementing similar concepts can be found in the literature, such as the shifted POD,[38] where snapshots are shifted based on identified transport velocities, the method of Nair and Balajewicz,[39] where snapshots are transported based on parameter dependent transport fields defined in the training phase, and the method of Karatzas et al,[40] where problem specific transformations are used to transport snapshots to domains of varying geometry.

3.2.4 | Interpolation

Interpolation in parametrized reduced order modeling can be used both in the lower dimensional basis[16] or in the reduced system matrices level.[18,19,21] While the first alternative would fail to transfer the discontinuity, in a similar manner to the direct application of POD, the second alternative could be employed to interpolate the system matrices of the reduced model. A limitation of this approach is that for large numbers of parameters, which might be required to describe complex crack geometries, the number of data points needed to accurately interpolate the system matrices might become unfeasible. Moreover, this approach is mainly applicable to linear problems.

3.2.5 | Proposed approach

Since in the present work we aim at developing methods exploiting information obtained from pre-computations carried out for different crack configurations, we focus on methods of the last three categories. Regarding mesh morphing and mapping of modes, both methods rely on some mapping between different crack configurations. In fact, mesh morphing techniques can be seen as such a mapping. The main difference lies in the fact that, in the mode mapping case, once the domain and modes are mapped to the target configuration, the modes are interpolated so that they refer to the original configuration. In our approach, mesh morphing is preferred since it avoids interpolation of the modes among different meshes, which introduces additional complexity and errors. Mesh distortion problems associated with this approach are mitigated through a technique introduced in the following section. Of course, the resulting technique, in contrast to methods relying on decomposing the solution, can only deal with the specific crack topologies used in the training phase. For instance, a reduced model trained for edge cracks at one specific boundary can not be used for interior cracks or edge cracks at a different boundary. Such cases can only be handled by training separate reduced models.

Based on the above, standard finite elements could be used instead of XFEM since different crack configurations can be represented using morphed meshes rather than enrichment. Nevertheless, XFEM can still be advantageous since it allows the use of different reference configurations without requiring remeshing. This can be used either to automate the selection of reference configuration, or to allow the use of multiple reference configurations in case the required range of crack configurations cannot be covered by mesh morphing alone.

Finally, as described in more detail in Section 5.2.1, interpolation of the system matrices can also be employed in conjunction to our approach to further increase efficiency.

4 | MESH MORPHING BASED ON MODIFIED INVERSE DISTANCE WEIGHTING

For the mesh morphing problem, we start with the initial or reference domain Ω, with boundary Γ and a crack Γ_c, which we term the reference crack. We assume that a mesh is defined in this domain consisting of N_h nodes. The objective is to deform points of the mesh, such that the reference crack coincides with a target crack $\overline{\Gamma}_c$, while preserving the boundaries of the domain. In the present case, preservation of the boundaries is accomplished by constraining the respective points to only move along the boundary, rather than completely preventing them from moving. This is necessary for cases where the reference crack is translated along a boundary, while it can also improve the quality of the deformed mesh in the case of interior cracks.

In general, several mesh morphing methods can be found in the literature, for instance discrete minimal surfaces[41] as in the work of Galland et al,[29] free form deformation,[42] radial basis function interpolation,[43] inverse distance weighting (IDW),[44] as well as physical analog methods such as the ones proposed by Farhat et al,[45,46] relying on torsional springs.

The development of our mesh morphing method is mainly motivated by two requirements posed by the particularities of the physical problem:

- The method should be flexible with respect to the deformation of points on the boundaries of the domain considered.
- The method should be efficient to allow its application in the online phase.

In most of the available techniques, mesh morphing is controlled through a set of points C, referred to as control points, where deformation is prescribed. In order to provide the flexibility required to tackle deformation at the boundaries, we follow a hierarchical procedure where topological entities, such as points, curves, and surfaces, are defined. Subsequently, deformations for each entity are defined in increasing order of topological dimensions using as control points all points belonging to entities of lower dimensions, as well as the initial control points. Then, provided that points belonging to entities of lower dimensions can be constrained to deform only within the boundaries of these entities, the deformations prescribed for entities of higher dimensions will not alter their boundaries.

As an example, a two-dimensional solid is a surface of topological dimension 2, while its boundaries can be defined as curves of topological dimension 1. In the case of open curves, the endpoints are also defined as topological entities of dimension 0. Then, the deformation of points would be computed first, followed by deformations of the curves and finally the deformations of the surface. For open curves, the deformations of the endpoints in conjunction to the deformation of

the initial control points would be used to drive the morphing process. Similarly, in order to morph the mesh in the interior of the domain, all points belonging to the boundaries would be used as control points and their deformations, computed in the previous stage, as well as the deformations of the original control points would be used to drive the process.

Morphing at each stage of the above hierarchical procedure is performed using a novel approach, based on the IDW method.[44] An important feature of the IDW method is the possibility to compute deformations of points individually, thus allowing to morph only parts of the mesh if desirable. This feature, as will be shown in the following sections, can be exploited to substantially increase the overall efficiency of the proposed framework. Additionally, through the proposed modifications, the number of operations required to compute the deformation of each point can be rendered independent of the number of control points, further increasing the efficiency of the method.

4.1 | Inverse distance weighting

In the IDW method, the deformed location of each point is obtained by adding a deformation to its initial coordinates:

$$\bar{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{s}(\boldsymbol{x}), \tag{20}$$

where $\bar{\boldsymbol{x}}$ are the final coordinates of the point, $\boldsymbol{x}$ are its initial coordinates, and $\boldsymbol{s}$ is the deformation, defined as a weighted sum of the deformations of the control points:

$$\boldsymbol{s}(\boldsymbol{x}) = \sum_{i \in C} w_i(\boldsymbol{x}) \boldsymbol{s}_i, \tag{21}$$

with the weights of each point being functions of its distance from the control points:

$$w_i(\boldsymbol{x}) = \begin{cases} \frac{\|\boldsymbol{x}-\boldsymbol{c}_i\|^{-p}}{\sum_{j \in C} \|\boldsymbol{x}-\boldsymbol{c}_j\|^{-p}} & \text{if} \quad \boldsymbol{x} \neq \boldsymbol{c}_i \\ 1 & \text{if} \quad \boldsymbol{x} \equiv \boldsymbol{c}_i \end{cases}, \tag{22}$$

where $\|\square\|$ is the Euclidean norm and p a positive integer, typically $p = 2$.

Assuming that a mesh is defined in the domain, the deformation of all nodes can be written compactly as:

$$\hat{\boldsymbol{s}} = \mathbf{W} \boldsymbol{s}_c, \tag{23}$$

where $\hat{\boldsymbol{s}} \in \mathbb{R}^{N_h \times N_d}$ is the matrix of nodal deformations, $\mathbf{W} \in \mathbb{R}^{N_h \times N_c}$ is a matrix containing the weights of Equation (22) for all nodes and control points, and $\boldsymbol{s}_c \in \mathbb{R}^{N_c \times N_d}$ is a matrix containing the deformation of all the control points. In the above definitions, N_d is the number of spatial dimensions (in the present work $N_d = 2$) and N_c is the number of control points.

4.2 | The proposed approach

One drawback of the IDW approach is that, typically and in order to accurately represent the deformation of surfaces or curves, all of their nodes are used as control points. Then, the number of operations required to form matrix $\mathbf{W}$ and perform the product of Equation (23) is $\mathcal{O}(N_h N_c)$. Since, in our approach, all points of the boundary are defined as control points, and assuming that the number of these points is approximately $\mathcal{O}(\sqrt{N_h})$ for the two dimensional and $\mathcal{O}(N_h^{2/3})$ for the three dimensional case, the required operations are $\mathcal{O}(N_h^{3/2})$ and $\mathcal{O}(N_h^{5/2})$, respectively. In the work of Luke et al,[47] computational complexity was reduced by approximating the weighted sum by a sum over a smaller number of points, while in Ballarin et al,[48] a selection process was employed to reduce the original number of control points, as well as a procedure based on POD to further reduce the online cost of the method.

Herein, we propose a variant of the original method, which allows to both guarantee the accurate representation of the control surfaces/curves and reduce the computational cost for all phases while preserving the simplicity of the original method. The development of our variant is motivated by the observation that, in the original method, even though the

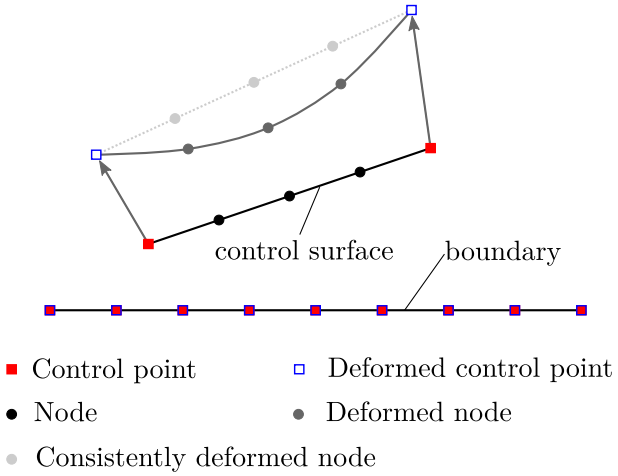


FIGURE 2 Deformation of a control surface close to a boundary [Colour figure can be viewed at wileyonlinelibrary.com]

objective is to conform the mesh to the deformed control surfaces or curves, weights are assigned based on the distance from the control points. As a result, for points lying on a control surface, weights corresponding to other control surfaces do not assume a value of zero leading to oscillations in the deformed surface. The problem is illustrated through a simple example in Figure 2, where the deformation of a linear control curve in the vicinity of a boundary is considered. Even though both the original and the deformed curve can be described exactly by two control points, with IDW all of the nodes lying on the control curve have to be defined as control points, otherwise distortion of the deformed control curve occurs, as illustrated. A simple modification to overcome the above issue consists of weighting deformations with distances from control surfaces/curves rather than distances from control points. Then, points lying on the control surfaces are assigned a value of unity for the weight corresponding to the surface itself and a value of zero for all other surfaces, thus preserving deformed control surfaces in any case.

In what follows, we will use the term control surfaces to describe both surfaces and curves since they will be treated in an equivalent way. To formulate our approach, we define control surfaces parametrically using the control points:

$$\bar{\boldsymbol{x}}(\boldsymbol{\xi}) = \sum_{i \in C} N_i(\boldsymbol{\xi}) \boldsymbol{c}_i, \tag{24}$$

where $\bar{\boldsymbol{x}}$ are the coordinates of points on the surface, N_i are basis functions and $\boldsymbol{\xi}$ is a vector of parameters. In case the domain is discretized, the basis functions used for discretization can also be used to represent the control surfaces. Similarly, the deformation of any point on the control surface can be obtained as:

$$\bar{\boldsymbol{s}}(\boldsymbol{\xi}) = \sum_{i \in C} N_i(\boldsymbol{\xi}) \boldsymbol{s}_i. \tag{25}$$

The distance of any point from a control surface Γ_i can be obtained by means of closest point projection as:

$$d_i(\boldsymbol{x}) = \min_{\bar{\boldsymbol{x}}_i \in \Gamma_i} \|\boldsymbol{x} - \bar{\boldsymbol{x}}_i\|, \tag{26}$$

where $\bar{\boldsymbol{x}}_i$ is the projection of point $\boldsymbol{x}$ on the surface. For the two dimensional examples tested in the present work, these projections are preformed in a brute force fashion by computing all distances from each point of the mesh to each point of the boundary. For a potential three dimensional extension, these projections can be performed efficiently by employing tree structures, as for instance in the work of Guezlec.[49] The deformation of point $\bar{\boldsymbol{x}}_i$ can also be obtained from Equation (25):

$$\bar{\boldsymbol{s}}_i = \boldsymbol{s}(\bar{\boldsymbol{\xi}}_i), \tag{27}$$

where $\bar{\boldsymbol{\xi}}_i$ are the parameter values corresponding to point $\bar{\boldsymbol{x}}_i$.

With the above definitions in place, the deformation of a point in the mesh can be defined as a weighted average of the deformations of its projections on the control surfaces:

$$\boldsymbol{s}(\boldsymbol{x}) = \sum_{i \in S} w_i(\boldsymbol{x}) \bar{\boldsymbol{s}}_i, \tag{28}$$

where S is the set of all control surfaces. Weights for each surface are defined similar to the original IDW method, replacing however distances from the control points with distances from the control surfaces:

$$w_i(\boldsymbol{x}) = \begin{cases} \frac{d_i(\boldsymbol{x})^{-p}}{\sum_{j \in S} d_j(\boldsymbol{x})^{-p}}, & \text{if} \quad \boldsymbol{x} \neq \boldsymbol{c} \\ 1, & \text{if} \quad \boldsymbol{x} \equiv \boldsymbol{c} \end{cases}. \tag{29}$$

With the modification introduced, the complexity associated with computing the deformation of each point is $\mathcal{O}(N_{\text{Surf}})$, with N_{Surf} the number of surfaces. Since a large number of control points would typically be required to define each control surface, the proposed modification significantly reduces the computational effort associated with the procedure.

4.3 | Treatment of boundaries

As already mentioned, the deformation of both surfaces and curves can be obtained using the proposed variant of the IDW method. However, a straightforward application of the method to curves describing boundaries of the domain would result in morphed meshes that do not preserve the initial geometry. Moreover, a distinction is made between open and closed curves, since endpoints of open curves can be considered as topological entities of lower dimension, thus their deformations should be determined prior to the deformations of interior points.

4.3.1 | Open boundaries

The shape of open curves can only be preserved by imposing a zero deformation at their endpoints, which can then be used as control points for morphing the remaining points. Regarding preservation of the shape in the interior of the curve, a further distinction is made between straight and curved boundaries, since in the former case a more efficient procedure can be used. In both cases, the deformation of the projections of points on the control surface are initially computed using Equations (25) and (27).

Straight boundaries

For the case of straight boundaries, points are projected on the control surfaces to obtain the initial deformation vectors $\boldsymbol{s}(\bar{\boldsymbol{x}}_i)$. Then a correction is performed by projecting these vectors on the surface:

$$\bar{\boldsymbol{s}}^c = (\boldsymbol{s}(\bar{\boldsymbol{x}}_i) \cdot \boldsymbol{t}_i)\boldsymbol{t}_i, \tag{30}$$

where $\bar{\boldsymbol{s}}^c$ is the corrected deformation vector, $\boldsymbol{s}(\bar{\boldsymbol{x}}_i)$ is the initially computed deformation vector, and $\boldsymbol{t}_i$ is the unit tangent vector of surface i. These corrections guarantee that the deformation vectors corresponding to each control surface will be parallel to the boundary.

Once the correction is applied, the final deformation of each point can be computed using Equation (28), with the corrected deformations. Since deformations of the end points are set to zero and deformations corresponding to the control surfaces are modified to remain parallel to the surface, the resulting deformed points will also lie on the original surface.

Curved boundaries

For curved boundaries, correcting the direction of the deformation corresponding to the control surface does not guarantee that the deformed points will still lie on the boundary since the tangent of the boundary is not constant. Thus, a slightly different procedure is used, where an initial deformation is obtained using Equation (28) and the deformations

of the projections of points on the control surface. The new location of the points is obtained as:

$$\boldsymbol{x}^0 = \boldsymbol{x} + \bar{\boldsymbol{s}}(\bar{\boldsymbol{x}}_i), \tag{31}$$

where $\boldsymbol{x}^0$ are the deformed points, not lying on the boundary, and $\bar{\boldsymbol{s}}(\bar{\boldsymbol{x}}_i)$ is the initially computed deformation. Subsequently, these points are projected on the initial boundary, thus restoring its shape, and the corrected deformation is defined as the difference between this projection and the initial points:

$$\boldsymbol{s}(\boldsymbol{x}) = \bar{\boldsymbol{x}}_i^0 - \boldsymbol{x}, \tag{32}$$

where $\bar{\boldsymbol{x}}_i^0$ is the projection of point $\boldsymbol{x}^0$ on the boundary i.

4.3.2 | Closed boundaries

In closed curves, the shape can be preserved without imposing any additional deformations, thus morphing is only driven by the deformation of the control surfaces. A simple way to deform points on these boundaries consists of simply translating them along the curve based on the deformation of some control point. To this end, a parametrization based on the length along the curve starting from some randomly selected reference point is introduced. The parameter value is then defined for each point as the sum of the lengths of all segments between this point and the reference point.

Subsequently, the point closest to the control surface is selected as a control point and it is assigned a deformation equal to:

$$\bar{\boldsymbol{s}}^c = w_{cc}\boldsymbol{s}(\bar{\boldsymbol{x}}_i), \tag{33}$$

where $\boldsymbol{s}(\bar{\boldsymbol{x}}_i)$ is the deformation of projection of the point on the control surface and w_{cc} is a weight defined as:

$$w_{cc} = \begin{cases} \frac{d_{\min}^{-2}}{d_{\min}^{-2} + d_c^{-2}}, & \text{if} \quad d_{\min} \neq 0 \\ 1, & \text{if} \quad d_{\min} = 0 \end{cases}, \tag{34}$$

with $d_{\min}$ the minimum distance from the point to the control surface and d_c the distance from the closest to the furthest point in the curve. The introduction of this weight, renders the deformation of the control point equivalent to a weighted average of the deformation of its projection on the control surface and the deformation of a fictitious control point, whose distance from the initial control point is d_c and whose displacement is zero. Using this definition, the deformation of the control point decreases as the distance from the control surface increases.

To ensure that the boundary is preserved, the deformed control point is projected on the boundary and the value of the parameter corresponding to the projected point is computed. Finally, the difference between the values of the parameter of the original and deformed point is used to deform all other points. This deformation is performed by interpolating the location of the deformed points on the original discretization and then transforming to the global coordinate system.

4.4 | Tuning the mapping

In order to add flexibility to the method and to allow to adapt the morphing process to specific geometries, we slightly modify the expression of Equation (29) for the weights as follows:

$$w_i(\boldsymbol{x}) = \begin{cases} \frac{(\alpha_i d_i(\boldsymbol{x}))^{-p_i}}{\sum_{j \in S} (\alpha_j d_j(\boldsymbol{x}))^{-p_j}}, & \text{if} \quad \boldsymbol{x} \neq \boldsymbol{c} \\ 1, & \text{if} \quad \boldsymbol{x} \equiv \boldsymbol{c} \end{cases}, \tag{35}$$

where p_i is the value of the exponent, defined individually for each surface, and α_i is a scaling factor used for the distance from each surface.

Then the exponents and scaling factors corresponding to different surfaces can be tuned to optimize the quality of the resulting morphed meshes. In the present work, rather than performing this process manually, we employ an optimization algorithm to minimize a mesh quality metric over a set of representative deformed mesh configurations. As a mesh quality metric we employ the size-shape metric, introduced in the work of Knupp,[50] and we define the optimization problem as:

$$\begin{aligned} &\textit{Find}\ \ \boldsymbol{\theta}\ \textit{such that} \\ &\mathcal{F}(\boldsymbol{\theta}) \rightarrow \min, \end{aligned} \tag{36}$$

where $\boldsymbol{\theta}$ is a vector containing all of the above parameters (p_i, α_i) and $\mathcal{F}$ is the objective function defined as:

$$\mathcal{F} = -\min_{i \in C_t} f_{ss}^i, \tag{37}$$

with C_t a set of training configurations and f_{ss}^i the value of the mesh quality metric for each configuration, defined as the minimum over all elements of the mesh, of the change in mesh quality. Similarly to Luke at al[47] and De Boer,[43] the change in mesh quality can be obtained as:

$$f_{ss}^i = \min_{e \in \mathcal{E}} \frac{\min\limits_{0}(f_e^0, f_e^i)}{\max\limits_{e}(f, f_e^i)}. \tag{38}$$

In the above definition, $\mathcal{E}$ is the set of all elements in the mesh, f_e^0 is the value of the size-shape metric of Knupp[50] evaluated for element e in the initial mesh, while f_e^i is the value of the same metric evaluated at element e in the mesh corresponding to configuration i. Provided that the training set contains a sufficient number of configurations, a positive value of the fitness function should guarantee that no degenerate elements will be present in the morphed meshes. Furthermore, since the initial value of the mesh quality metric is also used, the process should reduce the deviation from the initial mesh which is a desirable feature as well.

For the solution of the problem of Equation (36), we employ the covariance matrix adaptation evolution strategy (CMAES)[51,52] due to its excellent performance in non-convex highly nonlinear global optimization problems. It should be mentioned that several alternatives exist, some of which might provide improved performance, however a comparison would exceed the purposes of the present work.

During the optimization process, the repeated evaluation of the fitness function for all the configurations in the training set is required, which in turn involves morphing the mesh with varying sets of parameters and evaluating the mesh quality index. The mesh morphing part can be performed efficiently using the proposed method. In fact, since the reference configuration remains constant throughout the process, all the required distances can be pre-computed, leaving only the computation of the weights and the actual mesh morphing to be computed for each configuration. Regarding the mesh quality metric, evaluation for a single configuration is also relatively cheap computationally, since the operations required are approximately $\mathcal{O}(N_e)$, where e is the number of elements in the mesh. Nevertheless, if the number of training configurations is large, which can occur in cases where several parameters are used to describe the crack geometry, then the whole process can quickly become unfeasible. To remedy this issue, we start by making the following assumption: the configurations causing the most severe mesh distortion are the ones in which the target crack lies the furthest away from the reference crack. Then, from a larger pool of training configurations, we select a subset, leading to the largest values for some distance metric from the reference crack. The above assumption is based on intuition and confirmed by numerical results. Nevertheless, to ensure that the final choice of parameters will not result in degenerate elements, once the algorithm has converged, the mesh quality metric is evaluated for the whole set of training configurations.

As a further means of automating the mesh morphing process, we automatically select as a reference configuration the one that minimizes the distance to all other configurations in the training set. A choice consistent with the assumption made in the previous paragraph.

5 | SAMPLING AND PROJECTION INTO THE LOWER DIMENSIONAL SUBSPACE

5.1 | Sampling

The techniques described in the previous subsection allow to represent a large array of target crack geometries using a single reference configuration. However, apart from the geometry, also the solution for these target cracks needs to be properly represented by the ROM. To this end, we employ a greedy sampling procedure[53] to effectively select configurations that minimize the error of the reduced model. During this procedure, an initial reduced basis, and the corresponding ROM, are created, based on some specific realization of the model, in the present case the reference configuration, while a large pool of parameter sets is generated. Subsequently, the reduced model created is solved for all the generated configurations and the solution corresponding to the largest error, based on some a posteriori estimator, is added to the lower dimensional space. The process is repeated until the maximum error is reduced below a predefined threshold. In our case, we create the initial reduced model based on the reference configuration and generate a pool of crack configurations through Latin hypercube sampling. Since, in the present work, only static, linear elastic problems are dealt with, it is still feasible to solve the full model for all the generated configurations. Thus the true error, defined as the L_2 norm of the difference between the reduced and FOM solutions, is used instead of an estimate during the greedy process.

5.2 | Projection into the lower dimensional subspace

Once a lower dimensional basis has been constructed, the projection of Equation (15) should be performed. This projection should be performed for different values of the parameters during the online phase of the method. However, the complexity of the operation would still depend on the size of the full problem N, thus only allowing for minor computational gains. To perform this projection efficiently, we employ two alternative strategies, which can also be used in a complementary fashion.

5.2.1 | Hyper reduction

In the first strategy, we employ a technique commonly used in nonlinear problems and referred to as hyper reduction. Hyper reduction methods approximate the projected matrices by selecting a subset of elements or nodes where the corresponding quantities are evaluated. Since this subset is typically of much smaller size than the full set of elements or nodes, substantial computational gains can be obtained. From the several methods available in the literature, such as the method introduced in the work of Ryckelynck,[22] discrete empirical interpolation (DEIM),[54] Gauss Newton with approximated tensors (GNAT),[55] and energy conserving mesh sampling and weighting (ECSW),[56,57] we select the latter one due to the fact that it has been developed for nonlinear dynamic finite element models and is thus closer to our model problem.

Assuming that the full response of the system is not required at every evaluation, the efficiency of the method can be further increased, during the online phase, by morphing only elements selected by the hyper reduction method plus any additional elements required for post processing. This is possible since the proposed mesh morphing method allows the independent deformation of individual nodes. As a result, the computational time associated with mesh morphing also scales with the number of selected nodes, rather than the much larger number of total nodes.

5.2.2 | Interpolation of reduced order system matrices

In the second alternative, the projection is essentially avoided by pre computing the reduced order system matrices of Equation (16a) at a set of points in the parameter space and employing some form of interpolation to obtain their values at points not belonging to this set.[18,20] Since a single global basis is used, all reduced matrices refer to the same coordinates, and the interpolation can be performed directly as in the work of Degroote et al[19] without requiring any additional transformation as in Amsallem et al.[21] For interpolating the reduced order stiffness matrix, we employ the method proposed in the work of Amsallem et al,[21] where the Cholesky factors of the matrices are interpolated to ensure that the resulting matrix is symmetric and positive definite. For the reduced external force vector, we employ direct interpolation since it

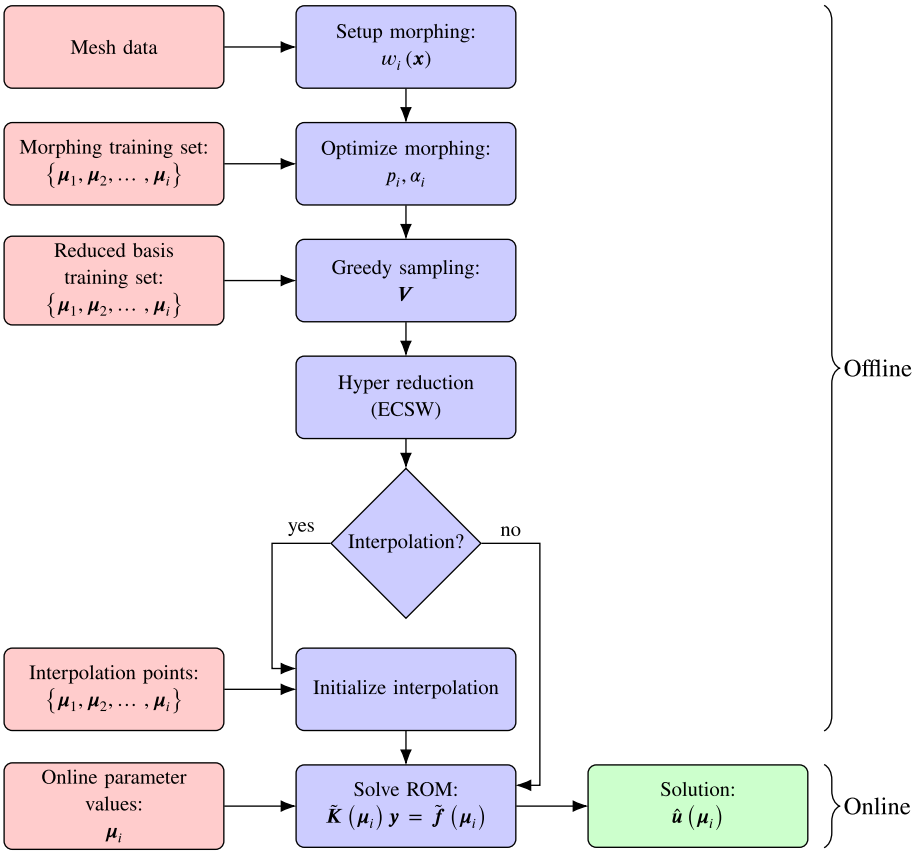


FIGURE 3 Main steps of the proposed approach [Colour figure can be viewed at wileyonlinelibrary.com]

does not possess any properties that should be preserved during interpolation. As an interpolation method, we employ splines as proposed in the work of Degroote et al.[19]

This approach can be very efficient computationally, while also minimizing interaction with the FOM. Nevertheless, in order to maintain accuracy in the problem studied, the parameter space should be sampled sufficiently, which for higher dimensions can lead to a very large number of points, in which the full order matrices have to be computed and projected into the lower dimensional space. In order to accelerate these projections, we employ a combined strategy, where hyper reduction is performed as described in Subsection 5.2.1, and subsequently used to compute all the required reduced order matrices. All of the steps of the resulting method are summarized in the flowchart of Figure 3.

6 | NUMERICAL EXAMPLES

6.1 | Implementation details

In what follows, the proposed ROMs are verified with respect to the solution of the corresponding full-order problem. This is achieved by computing the L_2 error of the nodal displacements of all nodes for one of the load orientations considered. In the training phase, unless otherwise stated, the tolerance for the greedy process is given a value of 1% and the tolerance for the nonnegative least square (NNLS) algorithm used in the ECSW method is set to 10^{-4}.

For all tested examples, three different sets of crack configurations are generated: the first spans the parameter space uniformly and is used to optimize the parameters of the mesh morphing method used, the second is generated through Latin hypercube sampling and is used in the greedy sampling process, while the third, generated independently from the second, also through Latin hypercube sampling, is used for computing errors with respect to the FOM. For the interpolation based approach, the parameter space is sampled uniformly and the number of samples in each direction is manually adjusted to obtain error levels similar to the hyper reduction based approach.

Mesh morphing is performed by considering the reference crack as the control surface an deforming it to match the target crack. In order to prevent elements containing the crack tips from deforming excessively, the crack surface is

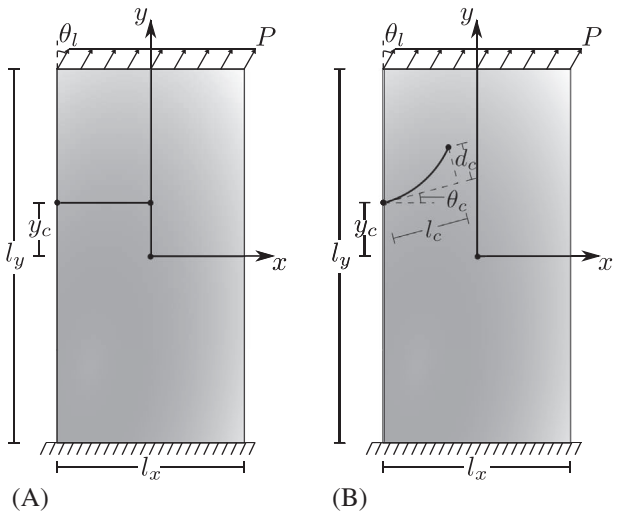


FIGURE 4 Geometry, loads, and crack parametrization for a tension/shear specimen with a straight (A) and curved (B) crack

extended in the direction of each crack tip by a segment of length 1.5 h, where h is the element size. In the optimization process described in Section 4.4, an initial exponent equal to 1.5 is used for all surfaces, while the initial scaling factors are given a value of 1 for all surfaces except the crack, which is assigned a value of 0.5. In order to restrict the search space for the CMAES algorithm upper and lower bounds are imposed on the above parameters. Thus, exponents are constrained to assume values in the interval $[1.0, 3.0]$ for the surface of the crack and $[0.5, 3.0]$ for all other surfaces. Similarly, scaling factors are limited in the interval $[0.2, 0.6]$ for the surface of the crack and $[0.6, 2.0]$ for all other surfaces. These values have been found to lead to relatively fast convergence of the optimization algorithm for all of the examples tested. Furthermore, different values are used for the surface corresponding to the crack in order to prevent excessive distortion of the enriched elements, which could result in numerical issues. For the CMAES algorithm, default settings were used and a maximum number of 4000 evaluations of the fitness function was used as a convergence criterion. Since the results of the optimization might vary between runs due to the stochastic nature of the algorithm, the obtained values for the exponents and scaling factors are provided for all of the examples.

Finally, for all of the examples, a Matlab implementation of the described methods was run on a workstation equipped with an Intel Xeon E3-1275 quad core processor, running at 3.80GHz, and 32GB of memory. Since the timings reported are often below a tenth of a second, they might not be very accurate, however they provide an indication of at least the order of magnitude of the provided speed up.

6.2 | Edge crack in a shear/tension specimen

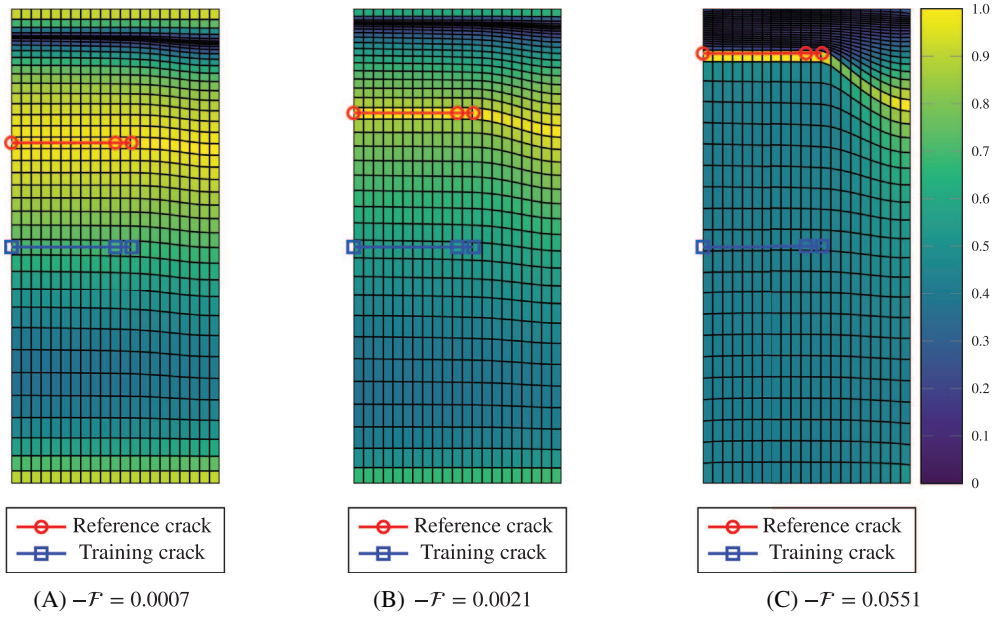
The first example consists of a cracked specimen under loads of varying orientation. As illustrated in Figure 4, two alternative scenarios are considered for the parametrization of the problem. In the first, a single parameter is considered representing the vertical location of an edge crack of fixed size and orientation. This configuration is employed to illustrate the effect of the optimization step in the procedure as well as the overall performance of the method in a simple setting. In the second scenario, a curved crack is considered, thus four parameters are used, representing the vertical location, orientation, size, and deviation from the direction of the tangent for a curved crack. In this configuration the effect of the addition of more parameters on the different components of the method can be studied.

The geometrical and material parameters are $l_x = 7$ units, $l_y = 14$ units, $E = 100$ units, and $\nu = 0.2$, where E and ν are the elastic modulus and Poisson's ratio, respectively. The load is given a unit value ($P = 1$ unit) and five different orientations: $\theta_l \in \{0, \pi/8, \pi/4, 3\pi/8\pi/2\}$. It should be mentioned that both in the present and following examples, load orientation is not treated as an additional parameter. Instead solutions for all orientations are obtained for each set of parameter values considered and used as snapshots. The initial mesh used is structured, consisting of 21×41 bilinear quadrilateral elements. The crack surface is represented as one linear segment in the first scenario and as 10 linear segments in the second.

TABLE 1 Shear/tension specimen with a straight crack described by a single parameter

	Uniform values: $y_c \in [-3.5, 3.5]$		Uniform values: $y_c \in [-4.5, 4.5]$		Optimized values: $y_c \in [-6.5, 6.5]$	
Surface	p_i	α_i	p_i	α_i	p_i	α_i
Bottom	2.0000	1.0000	1.5000	1.0000	1.0286	0.7753
Top	2.0000	1.0000	1.5000	1.0000	1.0285	0.7755
Left	2.0000	1.0000	1.5000	1.0000	1.0747	0.6909
Right	2.0000	1.0000	1.5000	1.0000	1.1970	0.7002
Crack	2.0000	0.5000	1.5000	0.5000	1.3313	0.4497

Note: Values of the exponents and scaling parameters used for each surface in the modified IDW method and widest parameter range possible for three different cases. In the first two cases, a uniform exponent of 2 and 1.5, respectively, is used for all surfaces and the default values for the scaling factors, while in the third case, all values are obtained through the optimization process described in Section 4.4.



(A) $-\mathcal{F} = 0.0007$ (B) $-\mathcal{F} = 0.0021$ (C) $-\mathcal{F} = 0.0551$

FIGURE 5 Shear/tension specimen with a straight crack described by a single parameter. Morphed meshes and mesh quality metric for the choices of exponents and scaling factors listed in Table 1. A, $-\mathcal{F} = 0.0007$. B, $-\mathcal{F} = 0.0021$. C, $-\mathcal{F} = 0.0551$

6.2.1 | Mesh morphing

Straight crack described by a single parameter

As a first test, we study the effect of the choice of exponents and scaling factors for the mesh morphing method on the mesh quality, as well as the possible range of parameters. To this end, we consider the first scenario and three cases: (a) a value for the exponents of Equation 35 equal to 2, as typically chosen in the literature, and default values for the scaling factors, (b) the initial value of the exponents used herein, equal to 1.5, and default values for the scaling factors, and (c) the values computed by the proposed optimization process with a training set of 11 cracks, uniformly spanning the search space. For all of aforementioned cases, the range of parameter y_c was altered to obtain the largest possible variation resulting in the ranges of Table 1. In Figure 5, the morphed meshes along with the mesh quality metric and the selected reference crack are plotted for each case, for the crack lying the furthest from the reference crack. From the above results, it is evident that the choice of exponents and scaling factors has an important effect on both the quality of the morphed meshes and the possible range of parameters; thus, optimization can significantly extend the range of possible configurations to be covered using a single reference crack.

Curved crack described by four parameters

For the second scenario, we only employ the optimization approach with a training set resulting from a uniform sampling of the parameter space using 11 values for the first, second, and fourth parameter and 5 for the third, leading to a set of

TABLE 2 Shear/tension specimen with a curved crack described by four parameters

	Optimized values	
Surface	p_i	α_i
Bottom	1.3471	1.5874
Top	1.3475	1.5870
Left	2.2886	1.4060
Right	1.4002	0.6058
Crack	1.7658	0.4258

Note: Values of the exponents and scaling factors used for each surface in the modified IDW method. These values are obtained as a result of the optimization process described in Section 4.4.

6655 configurations. The computed values of the exponents and scaling factors are given in Table 2, while the parameter ranges were set to: $y_c \in [-2, 2]$, $\theta_c \in [-2, 2]$, $l_c \in [3, 4]$, $d_c \in [-1, 1]$. As can be expected, the higher number of parameters inevitably restricts the range that can be achieved for each individual parameter without resulting in degenerate elements. Nevertheless, a relatively wide range can still be achieved for each parameter.

6.2.2 | Dimensionality reduction

Next, we study the effect of the number of parameters on the size of the resulting ROM in terms of the number of modes, number of elements, and number of points in the parameter space required for a certain level of accuracy.

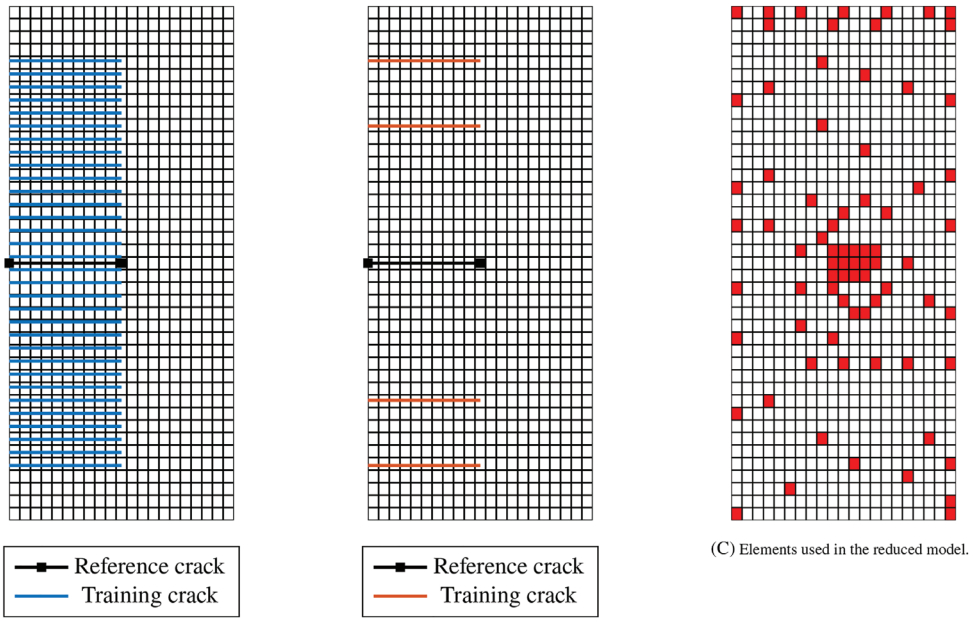
Straight crack described by a single parameter

Figure 6 summarizes the training and hyper reduction steps of the method. More specifically, in Figure 6A,B, the initial pool consisting of 64 training cracks and the 5 cracks selected by the greedy algorithm, resulting in 10 POD modes, are illustrated respectively. It can be easily noticed that, apart from the reference crack, which is used by default, the greedy algorithm selects the cracks lying the furthest away from the reference, as well as some lying in a slightly smaller distance, thus allowing to capture the difference in behavior exhibited by the solution as the crack moves along the edge of the specimen. Moreover, in Figure 6C, where the elements selected by the ECSW method are highlighted in the reference configuration, a concentration of selected elements around the crack tip is immediately noticeable. This is consistent with the stress concentration expected in the same area. In order to interpolate system matrices for this scenario, the sample space was sampled uniformly with 10 points. This relatively high number can be attributed to the fact that the location of the crack significantly impacts the response of the system. Therefore, in order to accurately capture this response over a large array of crack locations, a dense sampling of the system matrices is necessary. All of the above results are also summarized in Table 3.

Curved crack described by four parameters

The corresponding results for the second scenario are summarized in Figure 7. From Figure 7A,B, where the initial pool of 128 training cracks and the selected 12 cracks are illustrated, it is clear that a much larger number of training configurations is required to handle the increased number of parameters. Regarding the selected configurations, although it is harder to draw conclusions, it seems that the algorithm again tends to first select configurations further away from the reference. Similarly, in Figure 6C, it can be noticed that a larger number of elements is required, due to the higher number of configurations to be represented.

For interpolating system matrices, a uniform grid consisting of $9 \times 7 \times 7 \times 9$ points in the parameter space is required, leading to a total of 3969 combinations. The system matrices for these configurations are efficiently evaluated using the hyper reduction based ROM, nevertheless the very large number of parameter space points required could eventually constitute a limitation of the interpolation based approach for this type of problems. All of the above information is also summarized in Table 4.



(A) Initial pool of 32 training cracks. (B) 5 selected cracks. (C) Elements used in the reduced model.

FIGURE 6 Shear/tension specimen with a straight crack described by a single parameter. Cracks used for training and elements selected by the ECSW method. A, Initial pool of 32 training cracks. B, Five selected cracks. C, Elements used in the reduced model

TABLE 3 Shear/tension specimen with a straight crack described by a single parameter

	FOM	ROM1	ROM2
Size	1,920	10	10
Elements	861	56	—
Parameter space samples	—	—	10
Maximum error %	0.0000	1.6971	2.1538
Mean error %	0.0000	0.4885	0.9203
Solution time (s)	0.1271	0.0172	0.0014
Morphing time (s)	—	0.0029	—
Speedup	1.0000	6.1401	90.7857

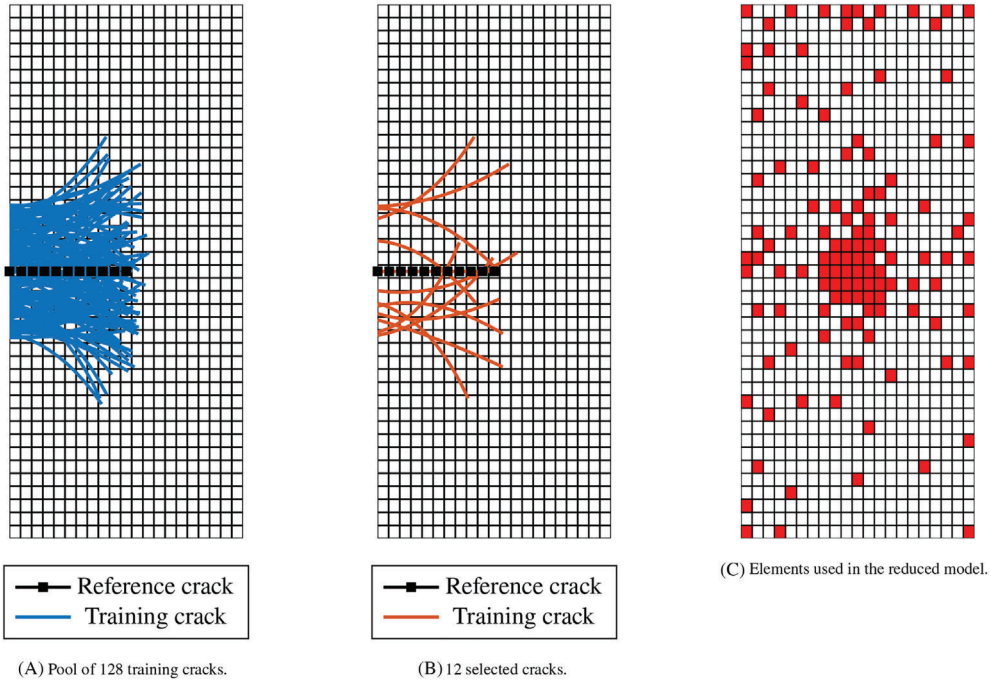
Note: Size, errors, and timings for all the models are considered.
Abbreviations: FOM, full order model; ROM1, reduced order model with hyper reduction; ROM2, reduced order model with matrix interpolation.

6.2.3 | Accuracy estimation

Straight crack described by a single parameter

To estimate the accuracy of the constructed ROMs, a set of 64 configurations is independently generated and errors are computed between the ROMs and the corresponding full models. The results are summarized in Table 3, where timings and other information regarding the size of all the models considered are also provided. Despite the small size of the original problem and the fact that the problem is linear and static, a substantial speedup can be achieved, especially by the interpolation based model, while keeping errors at a relatively low level compared to the full model. It should also be mentioned that the errors reported for the interpolation based approach are obtained using hyper reduction to compute the system matrices at the sampled points. In fact, it was verified that the difference in the resulting errors, compared to using the full model to compute these matrices, is insignificant. On the other hand, the difference in computational time is quite substantial as indicated by the difference in the number of elements between the FOM and ROMs.

As some additional indication of the effect of parameter variation on the solution, in Figure 8, the deformed specimen is illustrated for three different crack configurations, corresponding to different parameter values. It can be immediately noticed that, although solutions tend to differ widely, the ROM can successfully capture all different regimes with a relatively low number of modes.



(A) Pool of 128 training cracks. (B) 12 selected cracks. (C) Elements used in the reduced model.

FIGURE 7 Shear/tension specimen with a curved crack described by four parameters, cracks used for training and elements selected by the ECSW method. A, Pool of 128 training cracks. B, Twelve selected cracks. C, Elements used in the reduced model

	FOM	ROM1	ROM2
Size	1920	24	24
Elements	861	122	—
Parameter space samples	—	—	3,969 ($9 \times 7 \times 7 \times 9$)
Maximum error %	0.0000	0.7315	2.4100
Mean error %	0.0000	0.2870	0.5368
Solution time (s)	0.1293	0.0313	0.0085
Morphing time (s)	—	0.0048	—
Speedup	1.0000	3.5817	15.2118

Note: Size, errors, and timings for all the models considered.
Abbreviations: FOM, full order model; ROM1, reduced order model with hyper reduction; ROM2, reduced order model with matrix interpolation.

TABLE 4 Shear/tension specimen with a curved crack described by four parameters

Curved crack described by four parameters

In a similar fashion to the previous scenario, errors, timings, and model size information are provided in Table 4 for a set of 64 independently generated testing configurations. Again, low error levels can be achieved over a wide range of parameter values with a relatively small number of modes. The decreased speedup, compared to the first scenario, can be attributed to the increased number of elements required, in the case of the hyper reduction based model, and the increased number of modes, in the case of the interpolation based model.

6.3 | Interior crack in a unit square

The next example investigates the influence of the dimension of the proposed ROMs on the corresponding error levels, as well as the scaling of the computational time as the size of the FOM increases. To this end, we employ a unit square with an interior crack, as illustrated in Figure 9. The square is fixed at the lower side while a distributed load of variable orientation is applied to the upper end. The geometry of the crack is parametrized such that the first two parameters (x_c, y_c) correspond to the coordinates of its center point, while the remaining two (θ_c, l_c) represent its orientation and length.

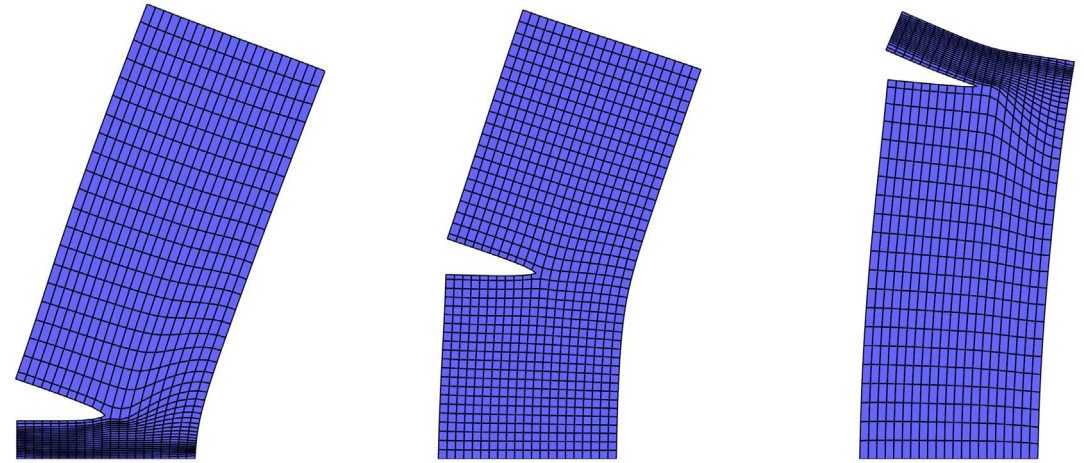
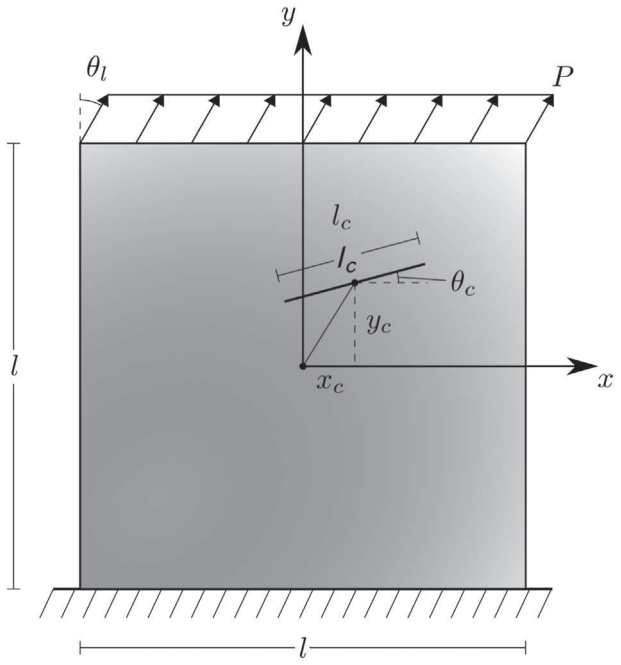


FIGURE 8 Shear/tension specimen with a straight crack described by a single parameter. Deformed mesh for three different crack locations. A, y_c=−6.5000 units. B, y_c=−0.9286 units. C, y_c=−6.5000 units [Colour figure can be viewed at wileyonlinelibrary.com]

FIGURE 9 Geometry, loads, and crack parametrization for a unit square with an interior crack



The geometrical and material parameters used are l=1 unit, E=100 units, and ν=0.2. The load is given a unit value (P=1 unit) and five different orientations: $\theta_l \in \{0, \pi/8, \pi/4, 3\pi/8\pi/2\}$. Parameters are given values in the following intervals: $x_c, y_c \in [-0.2, 0.2]$, $\theta_c \in [-\pi/4, \pi/4]$, and $l_c \in [0.15, 0.2]$. Finally, the domain is discretized using a structured mesh of 41×41 bilinear quadrilateral elements.

6.3.1 | Mesh morphing

To optimize the parameters used in our mesh morphing method, a training set of parameter values is generated by uniformly sampling the parameter space using 11 values for the first two parameters and 5 for the remaining two, resulting in a total of 3025 configurations. The automatically selected reference crack is depicted in Figure 10, along with the morphed mesh and mesh quality metric for two training cracks, while the values of the parameters obtained for the modified IDW method are given in Table 5.

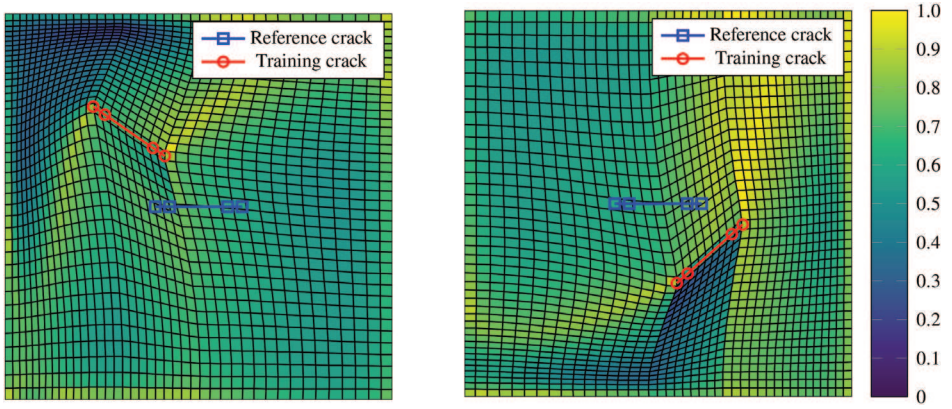


FIGURE 10 Unit square with an interior crack. Mesh quality metric and morphed meshes conforming to two cracks from the training set

TABLE 5 Unit square with an interior crack

Surface	p_i	α_i
Bottom	1.4477	0.9677
Top	1.4489	0.9685
Left	1.3875	0.7966
Right	1.3873	0.7965
Crack	1.3201	0.4455

Note: Values of the exponents and scaling parameters used for each surface in the modified IDW method. These values were obtained as a result of the optimization process described in Section 4.4.

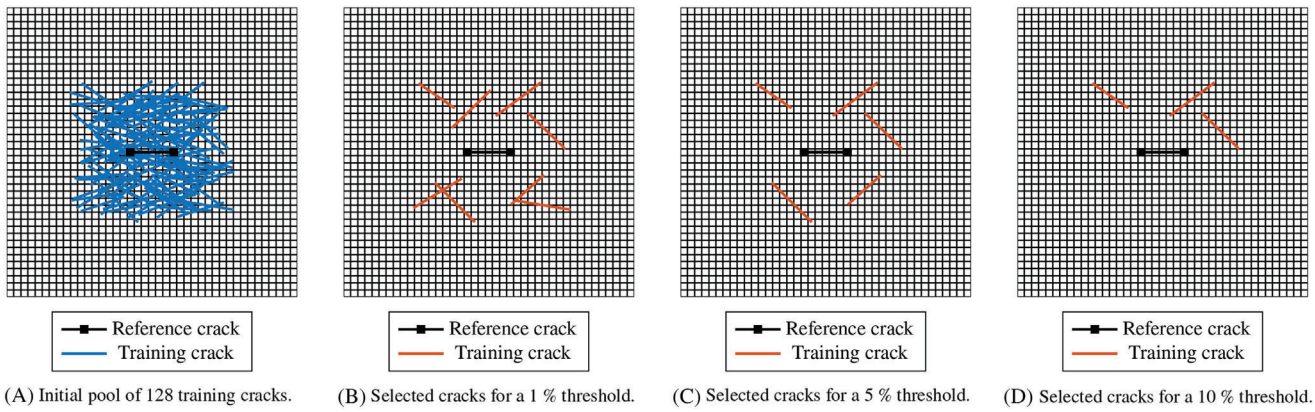
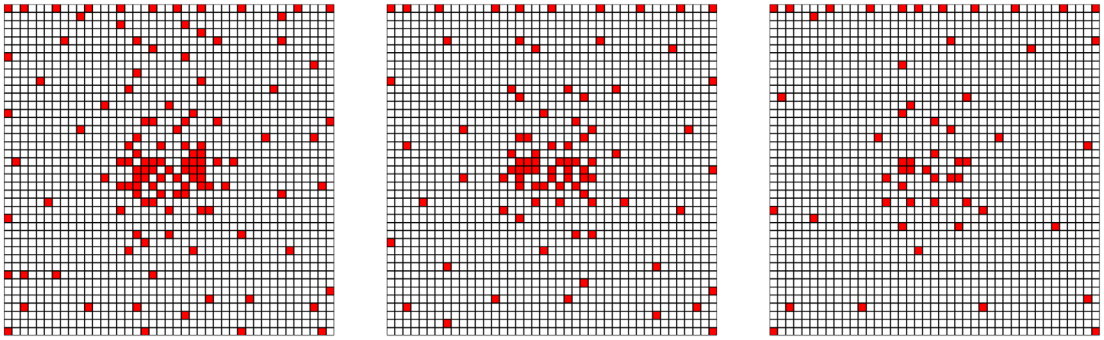


FIGURE 11 Unit square with an interior crack. Initial pool of training cracks and selected cracks for three different thresholds for the greedy sampling process. A, Initial pool of 128 traning cracks. B, Selected cracks of a 1% threshold. C, Selected cracks of a 5% threshold. D, Selected cracks of a 10% threshold

6.3.2 | Dimensionality reduction and accuracy

As a first test, ROMs are constructed using three different thresholds for the greedy sampling process with values 1%, 5%, and 10%, respectively. The results of the training step are illustrated in Figure 11 in terms of the selected crack configurations, and in Figure 12 in terms of the elements used in the ROM. As can be expected with an increasing accuracy, the selected number of cracks and elements increases. Moreover, similarly to the previous example, the greedy algorithm tends to first select cracks lying the furthest away from the reference crack, while the ECSW method tends to select elements where stress concentrations occur.



(A) Selected elements for a 1 % threshold in the greedy sampling process.

(B) Selected elements for a 5 % threshold in the greedy sampling process.

(C) Selected elements for a 10 % threshold in the greedy sampling process.

FIGURE 12 Unit square with an interior crack. Elements selected by the ECSW method for different numbers of selected training cracks. A, Selected elements of a 1% threshold in the greedy sampling process. B, Selected elements of a 5% threshold in the greedy sampling process. C, Selected elements of a 10% threshold in the greedy sampling process [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 6 Unit square with an interior crack

	FOM	ROM1-a	ROM1-b	ROM1-c	ROM2
Size	3,608	18	12	8	18
Elements	1,681	120	87	55	—
Parameter space samples	—	—	—	—	588 ($7 \times 7 \times 4 \times 3$)
Maximum error %	0.0000	1.0642	4.0799	5.3787	1.3021
Mean error %	0.0000	0.4694	1.0912	1.9666	0.5365
Solution time (s)	0.2317	0.0367	0.0270	0.0193	0.0054
Morphing time (s)	—	0.0034	0.0032	0.0031	—
Speedup	1.0000	5.7788	7.6722	10.3438	42.9074

Note: Size, errors, and timings for all the models considered.
Abbreviations: FOM, full order model; ROM1-a, reduced order model with hyper reduction and a 1% threshold for the greedy sampling process; ROM1-b, reduced order model with hyper reduction and a 5% threshold for the greedy sampling process; ROM1-c, reduced order model with hyper reduction and a 10% threshold for the greedy sampling process; ROM2, reduced order model with matrix interpolation constructed from model ROM1-a.

In Table 6, the sizes, errors, and timings are given for the ROMs constructed using the aforementioned thresholds, as well as a model constructed by interpolating the system matrices obtained using the most accurate of these models (the one with a threshold of 1%). The errors are computed for an independently generated set of 64 crack configurations. For the interpolation based model, the parameter space is sampled using a uniform $7 \times 7 \times 4 \times 3$ grid. The error levels obtained for all models are consistent with the threshold levels used, while computational times also decrease for the less accurate models. Similarly to the previous example, the model based on interpolation provides additional efficiency, requiring however a significant number of data points to cover the whole parameter space.

6.3.3 | Scaling of the ROM size

Next, we study the scaling of the computational cost associated with the proposed techniques by considering different levels of refinement for the FOM. More specifically, a series of structured meshes is employed consisting of $N_e \times N_e$ bilinear quadrilateral elements, with $N_e \in \{41, 81, 161, 321\}$. In Figure 13, the computational time associated with the FOM and ROM based on hyper reduction and interpolation respectively is plotted as a function of the size of the FOM. The computational time associated with both of the ROMs remains constant as the size of the FOM increases, which

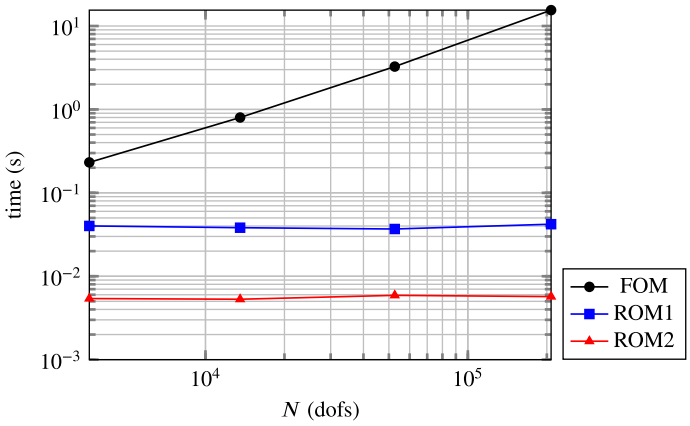


FIGURE 13 Unit square with an interior crack. Scaling of computational time associated with the full (FOM) and reduced order models based on hyper reduction (ROM1) and interpolation (ROM2) for increasing size (N) of the full order model. For the hyper reduction based model, mesh morphing time is also included in the reported timings [Colour figure can be viewed at wileyonlinelibrary.com]

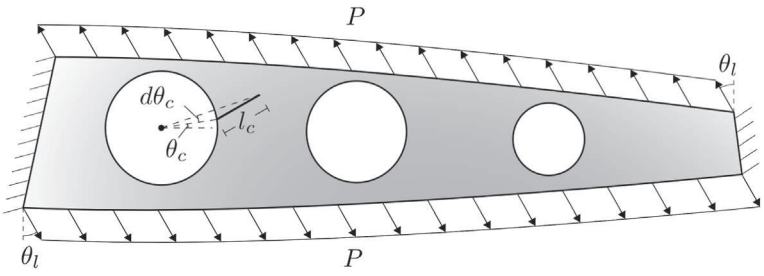


FIGURE 14 Geometry, loads, and crack parametrization for an aircraft wing rib

is the desired behavior. This is attributed to the fact that their complexity only depends on the number of POD modes used and the number of elements selected by the ECSW method, which for the same problem remain almost constant regardless of the level of refinement. Regarding the hyper reduction based model, the time needed for mesh morphing is also included in the computational time. However, it should be noted that, this constant computational time can only be achieved provided that the number of nodes/elements morphed remains constant, which in our case this is achieved by only morphing the nodes of elements selected by the ECSW method. In case the whole mesh had to be morphed, for instance for post processing purposes, then the computational cost would increase with the size of the full model.

6.4 | Rib of an aircraft wing

As a last example, we test the method in the complex geometry of the rib of an aircraft wing depicted in Figure 14. The boundary conditions, loading and crack locations, also shown in Figure 14 are not necessarily realistic; however, the example can still serve as a good demonstration for the capabilities of the method. The material parameters are E=100 units and ν=0.2. The load is given a unit value (P=1 unit) and five different orientations: $\theta_l \in \{0, \pi/8, \pi/4, 3\pi/8\pi/2\}$. The parameters are given values in the following intervals: $\theta_c \in [0, 0.15\pi]$, $l_c \in [40, 60]$, and $d\theta_c \in [-0.05\pi, 0.05\pi]$.

6.4.1 | Mesh morphing

The values of the parameters of the IDSW method, computed using the process described in Section 4.4, are given in Table 7. Compared to the previous examples, morphing in the present case is more time consuming as also illustrated in Table 8. This is attributed, apart from the increased model size, to the fact that, since the boundaries of the domain are curved, computation of their deformations requires projection of points on curves. As mentioned in Section 4.2, in our implementation, these projections are carried out in a brute force manner. Thus, if a more efficient method, such as the one proposed in the work of Guezlec[49] was employed, the required time could be substantially reduced.

TABLE 7 Aircraft wing rib

Surface	p_i	α_i
Bottom	1.6019	0.8510
Top	1.1523	1.7279
Left	1.8152	1.3582
Right	2.5865	0.7131
Hole 1	1.6425	1.1053
Hole 2	1.1302	1.3343
Hole 3	1.5557	0.7161
Crack	1.6396	0.4269

Note: Values of the exponents and scaling parameters used for each surface in the modified IDW method. These values were obtained as a result of the optimization process described in Section 4.4.

TABLE 8 Aircraft wing rib

	FOM	ROM1	ROM2
Size	13,912	24	24
Elements	6,526	258	—
Parameter space samples	—	—	250 ($10 \times 5 \times 5$)
Maximum error %	0.0000	1.7017	1.6345
Mean error %	0.0000	0.8676	0.9478
Solution time (s)	0.9901	0.0440	0.0073
Morphing time (s)	—	0.0274	—
Speedup	1.0000	13.8670	135.6301

Note: Size, errors, and timings for all the models considered.
Abbreviations: FOM, full order model; ROM1, reduced order model with hyper reduction; ROM2, reduced order model with matrix interpolation.

6.4.2 | Dimensionality reduction and accuracy

As in the previous examples, two reduced models are constructed, one employing hyper reduction and one based on interpolation. Results regarding the ROM sizes, errors, and timings are summarized in Table 8. Similar to the previous examples, the interpolation-based model outperforms the one employing hyper reduction. Furthermore, because of the increased size of the FOM, a significant speedup is achieved in both cases.

Finally, the effect of parameter variation on the solution is illustrated in Figure 15, where the deformed structure is shown for two different crack locations. Although the solution is affected by the presence of the crack globally, and not just in its vicinity, the ROM can accurately reflect this effect.

7 | CONCLUSIONS

A reduced order modeling methodology for cracked solids was presented, allowing to parametrize ROMs with respect to crack geometries and locations. The approach is based on mesh morphing, which can be carried out efficiently using a proposed variant of the IDW method. A global approach was employed for the construction of a lower dimensional subspace, into which the system matrices are projected. Two alternatives were employed to handle the projections in the online phase, the first employing hyper reduction by means of the ECSW method and the second relying on interpolation of the reduced system matrices. In order to accelerate the computation of the system matrices required for the

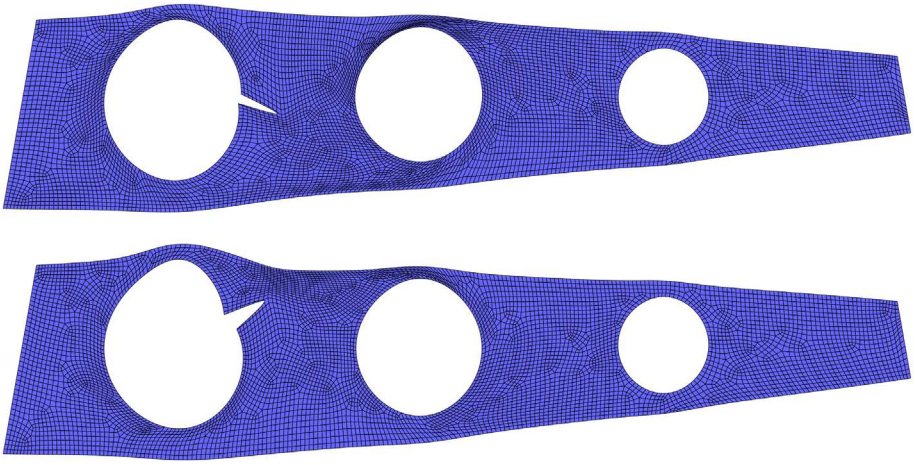


FIGURE 15 Aircraft wing rib. Deformed mesh for two crack configurations from the testing set [Colour figure can be viewed at wileyonlinelibrary.com]

interpolation-based approach, the hyper reduction based approach was employed. All of the different components of the method were tested through a series of numerical experiments involving different numbers of parameters, as well as varying levels of geometrical complexity. In general, the following conclusions can be drawn:

- Accurate results can be obtained over a wide range of parameter values using both approaches.
- Increasing the number of parameters describing the crack, limits, in general, the range of possible values for each of the parameters due to increased geometrical distortion.
- Albeit demonstrated on linear static problems, significant speedups are possible.
- The computational cost associated with mesh morphing, assembly, and solution of the hyper reduction based models can be kept almost constant with increasing refinement levels of the FOM, assuming that only the elements used in the reduced model are morphed.
- The hyper reduction based approach can provide highly accurate results while allowing a flexible treatment of large numbers of parameters. Additionally, this approach can be straightforwardly extended to nonlinear problems.
- As can be expected, for the linear problems considered herein, the interpolation-based approach significantly outperforms the hyper reduction method. Nevertheless, the large number of parameter space samples required for the type of problems studied might render the approach infeasible for increasing numbers of parameters. A possible remedy to this issue could be offered by adaptive sampling approaches.
- ROMs constructed using hyper reduction can be used to accelerate the computation of system matrices required for the construction of interpolation based models without significantly affecting the resulting error levels.

As already mentioned, a limitation of the proposed approach, related to mesh morphing, lies in the fact that only crack topologies used in the training phase can be solved for in the online phase, while the possible range of values assumed by each of the parameters can be limited due to geometrical distortion, especially for larger numbers of parameters. To mitigate these issues, the possibility of using multiple reference configurations in the training phase will be explored in the future. Regarding application to more complex problems, the extension to dynamics should be straightforward, while with the use of ECSW for hyper reduction, nonlinear problems could be handled. Of course, in the latter case, the usual limitations of POD for nonlinear problems would apply. Extension to three dimensional problems, should be possible, however the usual limitations of mesh morphing would apply, which for complicated geometries might reduce the applicability of the method. The methods could also be applied in crack propagation problems by employing a strategy, similar to the ones used in the work of Galland et al. [29], where the full order problem is periodically solved to enrich the reduced basis. Of course, limitations due to mesh morphing would also apply in that case. Finally, the use of adaptive sampling of the parameter space to decrease the number of data points required for the interpolation based approach might be necessary to allow for wider parameter ranges or larger numbers of parameters.

ACKNOWLEDGEMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No. 795917 "SiMAero, Simulation-Driven and On-line Condition Monitoring with Applications to Aerospace."

S.P.A. Bordas received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 811099 TWINNING Project DRIVEN for the University of Luxembourg and FNRS-FNR (INTER), Belgium-Luxembourg EnlightenIT - Multiscale modelling of lightweight metallic materials accounting for variability of geometrical and material properties.

CONFLICT OF INTEREST

No conflict of interest has been declared by the authors.

ORCID

Konstantinos Agathos https://orcid.org/0000-0002-9556-417X
Stéphane P. A. Bordas https://orcid.org/0000-0001-8634-7002

REFERENCES

1. Belytschko T, Black T. Elastic crack growth in finite elements with minimal remeshing. *Int J Numer Methods Eng*. 1999;45(5):601-620.
2. Moës N, Dolbow J, Belytschko T. A finite element method for crack growth without remeshing. *Int J Numer Methods Eng*. 1999;46(1):131-150.
3. Strouboulis T, Babuška I, Copps K. The design and analysis of the generalized finite element method. *Comput Methods Appl Mech Eng*. 2000;181(1-3):43-69.
4. Rabinovich D, Givoli D, Vigdergauz S. XFEM-based crack detection scheme using a genetic algorithm. *Int J Numer Methods Eng*. 2007;71(9):1051-1080.
5. Rabinovich D, Givoli D, Vigdergauz S. Crack identification by 'arrival time' using XFEM and a genetic algorithm. *Int J Numer Methods Eng*. 2009;77(3):337-359.
6. Jung J, Taciroglu E. Modeling and identification of an arbitrarily shaped scatterer using dynamic XFEM with cubic splines. *Comput Methods Appl Mech Eng*. 2014;278:101-118.
7. Waisman H, Chatzi E, Smyth AW. Detection and quantification of flaws in structures by the extended finite element method and genetic algorithms. *Int J Numer Methods Eng*. 2010;82(3):303-328.
8. Chatzi EN, Hiriyur B, Waisman H, Smyth AW. Experimental application and enhancement of the XFEM–GA algorithm for the detection of flaws in structures. *Comput Struct*. 2011;89(7-8):556-570.
9. Sun H, Waisman H, Betti R. Nondestructive identification of multiple flaws using XFEM and a topologically adapting artificial bee colony algorithm. *Int J Numer Methods Eng*. 2013;95(10):871-900.
10. Sun H, Waisman H, Betti R. A multiscale flaw detection algorithm based on XFEM. *Int J Numer Methods Eng*. 2014;100(7):477-503.
11. Yan G, Sun H, Waisman H. A guided Bayesian inference approach for detection of multiple flaws in structures using the extended finite element method. *Comput Struct*. 2015;152:27-44.
12. Agathos K, Chatzi E, Bordas SP. Multiple crack detection in 3D using a stable XFEM and global optimization. *Comput Mech*. 2018;62(4):835-852.
13. Zhang L, Yang G, Hu D, Han X. An approach based on level set method for void identification of continuum structure with time-domain dynamic response. *Appl Math Modell*. 2019;75:446-480.
14. Lumley JL. The structure of inhomogeneous turbulent flows. *Atmospheric* ***Turbulence*** *and* ***Radio Wave Propagation***; 1967.
15. Sirovich L. Turbulence and the dynamics of coherent structures. Part I: Coherent structures. *Q Appl Math*. 1987;45:561-571.
16. Amsallem D, Farhat C. Interpolation method for adapting reduced-order models and application to aeroelasticity. *AIAA J*. 2008;46(7):1803-1813.
17. Lieu T, Lesoinne M. Parameter adaptation of reduced order models for three-dimensional flutter analysis. 42nd AIAA Aerospace Sciences Meeting and Exhibit; 2004:888.
18. Amsallem D, Cortial J, Carlberg K, Farhat C. A method for interpolating on manifolds structural dynamics reduced-order models. *Int J Numer Methods Eng*. 2009;80(9):1241-1258.
19. Degroote J, Vierendeels J, Willcox K. Interpolation among reduced-order matrices to obtain parameterized models for design, optimization and probabilistic analysis. *Int J Numer Methods Fluids*. 2010;63(2):207-230.
20. Amsallem D, Farhat C. An online method for interpolating linear parametric reduced-order models. *SIAM J Sci Comput*. 2011;33(5):2169-2198.
21. Amsallem D, Tezaur R, Farhat C. Real-time solution of linear computational problems using databases of parametric reduced-order models with arbitrary underlying meshes. *J Comput Phys*. 2016;326:373-397.
22. Ryckelynck D. A priori hyperreduction method: an adaptive approach. *J Comput Phys*. 2005;202(1):346-366.

23. Huynh D, Patera A. Reduced basis approximation and a posteriori error estimation for stress intensity factors. *Int J Numer Methods Eng*. 2007;72(10):1219-1259.
24. Niroomandi S, Alfaro I, Gonzalez D, Cueto E, Chinesta F. Real-time simulation of surgery by reduced-order modeling and X-FEM techniques. *Int J Numer Methods Biomed Eng*. 2012;28(5):574-588.
25. O'Hara PJ, J. Hollkamp JJ. Modeling vibratory damage with reduced-order models and the generalized finite element method. *J Sound Vib*. 2014;333(24):6637-6650.
26. Wang X, Phlipot GP, Perez RA, Mignolet MP. Locally enhanced reduced order modeling for the nonlinear geometric response of structures with defects. *Int J Non-Linear Mech*. 2018;101:1-7.
27. Wang X, O'Hara P, Mignolet M, Hollkamp J. Reduced order modeling with local enrichment for the nonlinear geometric response of a cracked panel. *AIAA J*. 2018;57(1):421-436.
28. He Q, Chen JS, Marodon C. A decomposed subspace reduction for fracture mechanics based on the meshfree integrated singular basis function method. *Comput Mech*. 2019;63(3):593-614.
29. Galland F, Gravouil A, Malvesin E, Rochette M. A global model reduction approach for 3D fatigue crack growth with confined plasticity. *Comput Methods Appl Mech Eng*. 2011;200(5-8):699-716.
30. Kerfriden P, Gosselet P, Adhikari S, Bordas SPA. Bridging proper orthogonal decomposition methods and augmented Newton–Krylov algorithms: an adaptive model order reduction for highly nonlinear mechanical problems. *Comput Methods Appl Mech Eng*. 2011;200(5-8):850-866.
31. Kerfriden P, Passieux JC, Bordas SPA. Local/global model order reduction strategy for the simulation of quasi-brittle fracture. *Int J Numer Methods Eng*. 2012;89(2):154-179.
32. Kerfriden P, Goury O, Rabczuk T, Bordas SPA. A partitioned model order reduction approach to rationalise computational expenses in nonlinear fracture mechanics. *Comput Methods Appl Mech Eng*. 2013;256:169-188.
33. Moës N, Gravouil A, Belytschko T. Non-planar 3D crack growth by the extended finite element and level sets—Part I: mechanical model. *Int J Numer Methods Eng*. 2002;53(11):2549-2568.
34. Duflot M. A study of the representation of cracks with level sets. *Int J Numer Methods Eng*. 2007;70(11):1261-1302.
35. Zi G, Belytschko T. New crack-tip elements for XFEM and applications to cohesive cracks. *Int J Numer Methods Eng*. 2003;57(15):2221-2240.
36. Laborde P, Pommier J, Renard Y, Salaün M. High-order extended finite element method for cracked domains. *Int J Numer Methods Eng*. 2005;64(3):354-381.
37. Béchet É, Minnebo H, Moës N, Burgardt B. Improved implementation and robustness study of the X-FEM for stress analysis around cracks. *Int J Numer Methods Eng*. 2005;64(8):1033-1056.
38. Reiss J, Schulze P, Sesterhenn J, Mehrmann V. The shifted proper orthogonal decomposition: a mode decomposition for multiple transport phenomena. *SIAM J Sci Comput*. 2018;40(3):A1322-A1344.
39. Nair NJ, Balajewicz M. Transported snapshot model order reduction approach for parametric, steady-state fluid flows containing parameter-dependent shocks. *Int J Numer Methods Eng*. 2019;117(12):1234-1262.
40. Karatzas EN, Ballarin F, Rozza G. Projection-based reduced order models for a cut finite element method in parametrized domains; 2019. arXiv preprint arXiv:1901.03846.
41. Pinkall U, Polthier K. Computing discrete minimal surfaces and their conjugates. *Exp Math*. 1993;2(1):15-36.
42. Sederberg TW, Parry SR. Free-form deformation of solid geometric models. *ACM SIGGRAPH Comput Graph*. 1986;20(4):151-160.
43. De Boer A, Schoot VDM, Bijl H. Mesh deformation based on radial basis function interpolation. *Comput Struct*. 2007;85(11-14):784-795.
44. Shepard D. A two-dimensional interpolation function for irregularly-spaced data. Paper presented at: Proceedings of the 1968 23rd ACM National Conference; 1968:517-524.
45. Farhat C, Degand C, Koobus B, Lesoinne M. Torsional springs for two-dimensional dynamic unstructured fluid meshes. *Comput Methods Appl Mech Eng*. 1998;163(1-4):231-245.
46. Degand C, Farhat C. A three-dimensional torsional spring analogy method for unstructured dynamic meshes. *Comput Struct*. 2002;80(3-4):305-316.
47. Luke E, Collins E, Blades E. A fast mesh deformation method using explicit interpolation. *J Comput Phys*. 2012;231(2):586-601.
48. Ballarin F, D'Amario A, Perotto S, Rozza G. A POD-selective inverse distance weighting method for fast parametrized shape morphing. *Int J Numer Methods Eng*. 2019;117(8):860-884.
49. Guezlec A. "Meshsweeper": dynamic point-to-polygonal mesh distance and applications. *IEEE Trans Vis Comput Graph*. 2001;7(1):47-61.
50. Knupp PM. Algebraic mesh quality metrics for unstructured initial meshes. *Finite Elem Anal Des*. 2003;39(3):217-241.
51. Hansen N, Müller SD, Koumoutsakos P. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evol Comput*. 2003;11(1):1-18.
52. Hansen N, Kern S. Evaluating the CMA evolution strategy on multimodal test functions. Paper presented at: Proceedings of the International Conference on Parallel Problem Solving from Nature; 2004:282-291.
53. Haasdonk B, Ohlberger M. Reduced basis method for finite volume approximations of parametrized linear evolution equations. *ESAIM Math Modell Numer Anal*. 2008;42(2):277-302.
54. Chaturantabut S, Sorensen DC. Nonlinear model reduction via discrete empirical interpolation. *SIAM J Sci Comput*. 2010;32(5):2737-2764.
55. Carlberg K, Farhat C, Cortial J, Amsallem D. The GNAT method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows. *J Comput Phys*. 2013;242:623-647.

56. Farhat C, Avery P, Chapman T, Cortial J. Dimensional reduction of nonlinear finite element dynamic models with finite rotations and energy-based mesh sampling and weighting for computational efficiency. *Int J Numer Methods Eng*. 2014;98(9):625-662.
57. Farhat C, Chapman T, Avery P. Structure-preserving, stability, and accuracy properties of the energy-conserving sampling and weighting method for the hyper reduction of nonlinear finite element dynamic models. *Int J Numer Methods Eng*. 2015;102(5):1077-1110.

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section at the end of this article.

How to cite this article: Agathos K, Bordas SPA, Chatzi E. Parametrized reduced order modeling for cracked solids. *Int J Numer Methods Eng*. 2020;121:4537–4565. https://doi.org/10.1002/nme.6447